QUANTUM LEAP INNOVATIONS

March 4, 2011

LCDR Joseph Cohn
Program Officer, Code 341
Office of Naval Research
875 North Randolph Street
Arlington, VA 22203-1995
joseph.cohn@navy.mil

RE: Contract N00014-08-C-0036 - Final Report

Dear LCDR Cohn,

Work under contract N00014-08-C-0036 has been completed. Attached please find our Final Report and the SF-298 Report Documentation Page for:

Integrated Warfighter Biodefense Program (IWBP) – Phase 2

Covering the period October 2007 - December 2010

I will provide 2 sets of CD's (2 CD's per set) containing the software (CLIN 0001 – Data Items A0001, A0002 and A0003) developed for this contract via Federal Express.

Thank you for your assistance on the above noted program. Copies have been distributed as per the Contract Data Requirements List – Instructions for Distribution. Since the Final Report exceeds 30 pages, a hardcopy of the report will be mailed to the Director, Naval Research Lab as per the Instructions for Distribution.

Sincerely,

Frank T. Abbott
VP of Finance, CFO
fta@quantumleap.us

cc: Dr. Ganesh Vaidyanathan, Project Manager, Code 30, QLI gv@quantumleap.us
Administrative Contracting Officer – Stanley Brown, stanley.brown@dcma.mil
Director, Naval Research Lab, Attn Code 5596, reports@library.nrl.navy.mil
Defense Technical Information Center, tr@dtic.mil

**3 Innovation Way**
**Suite 100**
**Newark, DE 19711**
**phone 302.894.8000**
**fax 302.894.8001**
**www.QuantumLeap.us**

# REPORT DOCUMENTATION PAGE

*Form Approved*
**OMB No. 0704-0188**



| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From - To)* |
|---|---|---|
| 03-04-2011 | Final Report | Oct 2007 - Dec 2010 |

**4. TITLE AND SUBTITLE**
Integrated Warfighter Biodefense Program (IWBP) – Phase 2

**5a. CONTRACT NUMBER**
N00014-08-C-0036

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**
0603729N

**6. AUTHOR(S)**
Abbott, Franklin T.
Vaidyanathan, Ganesh

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Quantum Leap Innovations, Inc.
3 Innovation Way, Suite 100
Newark, DE 19711-5456

**8. PERFORMING ORGANIZATION REPORT NUMBER**
QLI-TR-11-002

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Office of Naval Research
ONR Code 341
875 North Randolph Street
Arlington, VA 22203-1995

**10. SPONSOR/MONITOR'S ACRONYM(S)**
ONR

**11. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**12. DISTRIBUTION AVAILABILITY STATEMENT**
Distribution A: Approved for public release; distribution is unlimited. 04 March 2011.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**
The aim of the Integrated Warfighter Biodefense Program (IWBP) is to develop innovative technology that can be deployed to prevent U.S. armed forces from becoming battle or non-battle casualties, and especially to reduce morbidity and mortality throughout the increasingly complex battlespace of current operations. The initial program was specifically linked to the operational requirements of end-users at US Northern Command (USNORTHCOM) and US Pacific Command (PACOM). In this summary of the Phase 1 work on IWBP we report the continued development of novel software that provides a simulation environment for modeling infectious diseases. The software was used to model diseases of geographical interest to USNORTHCOM and PACOM, to support a multi-national exercise (Cobra Gold '08), and supported US Marine Forces – Pacific (MARFORPAC) for Operation Caring Response (Cyclone Nargis – Myanmar). Additionally, the software was used to develop a capability to model shipboard disease in support of concerns about diseases impacting operational capability in future US Navy operations. And most recently, the software was used by USNORTHCOM to inform policy decisions surrounding the US government's coordinated response to Novel 2009 H1N1 influenza.

**15. SUBJECT TERMS**
Biological Defense, Emergency Management, Force Transformation, Situational Awareness, Course of Action Planning, Command and Control, Probabilistic Reasoning, Knowledge Discovery, Collaboration, Multi-Agent Systems, Agent-Based Modeling and Simulation

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | UU | 86 | Dr. Ganesh Vaidyanathan |
| Unclassified | Unclassified | Unclassified | | | 19b. TELEPONE NUMBER (*Include area code*) 302-894-8044 |

**Quantum Leap Innovations, Inc.**
**Delaware Technology Park**
**3 Innovation Way, Suite 100**
**Newark, DE 19711**

**QLI-TR-11-002**
**December 2010**

# Integrated Warfighter Biodefense Program (IWBP) – Phase 2

# Final Report

## ONR Contract N00014-08-C-0036

**Sponsored by the Office of Naval Research, Arlington, VA 22203**

## Abstract

The aim of the Integrated Warfighter Biodefense Program (IWBP) is to develop innovative technology that can be deployed to prevent U.S. armed forces from becoming battle or non-battle casualties, and especially to reduce morbidity and mortality throughout the increasingly complex battlespace of current operations. The initial program was specifically linked to the operational requirements of end-users at US Northern Command (USNORTHCOM) and US Pacific Command (PACOM). In this summary of the Phase 1 work on IWBP we report the continued development of novel software that provides a simulation environment for modeling infectious diseases. The software was used to model diseases of geographical interest to USNORTHCOM and PACOM, to support a multi-national exercise (Cobra Gold ’08), and supported US Marine Forces – Pacific (MARFORPAC) for Operation Caring Response (Cyclone Nargis – Myanmar). Additionally, the software was used to develop a capability to model shipboard disease in support of concerns about diseases impacting operational capability in future US Navy operations. And most recently, the software was used by USNORTHCOM to inform policy decisions surrounding the US government's coordinated response to Novel 2009 H1N1 influenza.

# Contents

## List of Figures

## List of Tables



## List of Appendices

# 1. Summary

### Executive Summary

This final technical report summarizes Quantum Leap Innovations' (QLI) accomplishments with the Integrated Warfighter Biodefense Program (IWBP) through the contract close date of December 31, 2010 on ONR Contract N00014-09-C-0033. QLI focused our final efforts on performing analysis on classified Office of Naval Research (ONR) Code 30 datasets and preparing final reports and paperwork.

### Summary of Accomplishments

Throughout the period of performance on this contract, QLI has been working on development and testing of components of the LeapWorks® Data Analytics platform. In particular, the LeapWorks Predictive Analytics component (formerly named Flexscape™) provides the capability to identify complex relationships inherent within a dataset. Models are built directly from the vast amounts of available data generating more accurate, useable and flexible models ready for advanced data analytics.

In addition to available ONR Code 30 data, QLI continues to actively seek appropriate datasets elsewhere to further validate the Flexscape technology. Examples of alternative datasets are described as use cases. QLI continues to investigate a variety of datasets including those in healthcare, pharmaceutical, financial, and consumer trends.

Phase 2 of the Integrated Warfighter Biodefense Program (IWBP) was funded under an FY2007 appropriation that has been managed through ONR Code 34. The program has been executed by Quantum Leap Innovations, Inc. (QLI) in a continuation of the Phase 1 program initiated in February 2007. The first phase of the program established a technology base for efforts that can leverage the full range of capabilities at QLI and provide technology innovations to ONR that can enhance US Navy operations.

# 2. Motivation from the Statement of Work

The initial demonstration of the IWBP technologies was focused on the threat from pandemic influenza to meet some immediate requirements defined by end-users as being of importance in implementing the pandemic response. Initial requirements were developed US Northern Command (USNORTHCOM) – the DoD lead for pandemic influenza planning and response. The initial focus on pandemic influenza enabled QLI to develop 'technology demonstrators' that support wider end-user reach and help focus the next phase of the program on force health protection and casualty prevention as defined by Code 34.

The IWBP continues to focus on previously identified targets for technology development including:

- Information Management, Modeling & Integration – Improving situational awareness, knowledge discovery, and knowledge management.

- Multi-Agent Framework & Intelligent Agents – Technology for multi-agent systems that can be designed, built, tested and deployed across distributed networks enabling automatic discovery and integration of services without a-priori knowledge of the details of the service.
- Medical Modeling & Situational Awareness – Improving medical situational awareness and responses to reduce morbidity and mortality.
- Flexible User Interfaces – Improving collaboration and knowledge interoperability across distributed enterprise environments.

These all support the focusing problem of infectious disease and are consistent with the requirements generated by USNORTHCOM.

# 3. Background

QLI is a technology company developing and deploying software products focused on the transformation of data into information and information into knowledge. Our software addresses problems that are characterized as being complex and dynamic. Our work is based on distributed computing, intelligent agents, and automated knowledge discovery technologies.

The work performed in Phase 2 of the IWBP focused on applying the resources of QLI and partner organizations to high-payoff deliverables that will save lives and help preserve a healthy and fit force. The program emphasizes intelligent computing technologies that address current and future evolving threats to our forces. By linking the program to current efforts that are underway at ONR and related organizations, the IWBP remains tightly coupled to end-user defined requirements.

# 4. Contract Activities

## 4.1 Information Management, Modeling and Integration

**Background:**

The work discussed in this section enables automated pattern discovery and predictive model building resulting in predictions that are inherently discrete or categorical in nature.

As a motivation for the ensuing summary of work on LeapWorks Data Analytics, it will be useful to revisit the basic tenets underlying the LeapWorks Pattern Discovery/Distributed Data Analysis (DDA) and Predictive Analytics/Distributed Model Fusion (DMF) capabilities that form the basis for this report and that are based upon the key concepts of Intelligent Data Management, Distributed Data Analysis and Distributed Model Fusion summarized in the Statement of Work for ONR Contract N00014-08-C-0036. In addition to enabling the detection and warning of a possible biological incident, the LeapWorks Data Analytics platform can facilitate the identification of data subsets that are relevant to a specific objective or set of objectives, as the basis for subsequent automated model building, hypothesis generation and simulations.

Pattern Discovery/DDA and Predictive Analytics/DMF represent two of the core modeling tasks within the Information Management, Modeling & Integration focus area that represents a significant component of the work that will be performed in Phase 2 of the IWBP. The motivation and rationale for the development of DDA and DMF is detailed in the contract Statement of Work:

> *"The Distributed Data Analysis (DDA) and Distributed Model Fusion (DMF) technologies will provide a system for remote modeling across distributed data sources where security, privacy or timeliness issues may preclude large scale data transport to a central data base. In the context of IBIS[1], the fragmented nature of the data across the different services makes it difficult to develop integrated knowledge that is distributed across the various data bases. Statistical relationships that are derived from a set of remote data sources may need to be combined to most accurately identify the likelihood and impact of an emerging health threat."*

LeapWorks Predictive Analytics automatically generates a population of models directly from data and subsequently combines these models to provide a consensus predictive capability. The discovery of a collection of informative patterns is the inherent defining characteristic of Distributed Data Analysis (or "DDA") and the subsequent generation and combining of individual models to arrive at a consensus prediction is the defining basis for Distributed Model Fusion (or "DMF"). Distributed Data Analysis followed by Distributed Model Fusion provides several benefits including:

(i) Scalability – Modeling a larger data environment by decomposing a larger data set into several smaller data subsets is significantly more efficient from a computational standpoint.

(ii) Flexibility – Applying the principle of Distributed Data Analysis across data from multiple data sources provides a natural mechanism for developing models directly at their resident data environments and subsequently using the models as significantly more compact "proxies" for the underlying data. In complex, dynamic data environments with potential bandwidth limitations, the ability to represent larger data sets with more compact models can provide significant advantages in the effective transformation of data to information to knowledge.

(iii) Robustness – Using multiple models to generate a consensus prediction can provide robustness to the prediction due to the redundancy inherent in the use of multiple predictions of a target variable. In addition, when new data is provided at a data source, only the "local" models associated with that data source need to be regenerated, without the need to rebuild the other models in the distributed data environment. This can make model maintenance significantly more robust.

## Pattern Discovery and Data Relevance:

### Motivation:

Traditionally, in the progression of data to information to knowledge, the role of data, though essential, has represented an early "pit stop" on the way towards knowledge discovery. Data is typically analyzed to identify important features of the data that can then be used to develop informative models or model components. A well-constructed model represents a compact description of the underlying data, and can be used to represent the data in the knowledge discovery process.

As the volume of data has increased over recent years, however, the amount of data has posed significant bottlenecks across the entire chain represented by the progression of data to information to knowledge. Data management has become increasingly complex and expensive, and the subsequent analysis of the data has suffered as well. In addition, the ability for humans to interpret the data in order to form testable theories or hypotheses becomes more difficult when confronted with vast amounts of data.

The ever increasing volume of data therefore places significant demands on data management, data storage and data utilization. The capability of "triaging" the data environment into data subsets that are relevant to specific applications can result in a data organization and filtering that can significantly enhance the subsequent extraction of knowledge from the data. Triaging data into "relevant" and "irrelevant" subsets can potentially enhance the value of the data to an enterprise as the information is now concentrated in the relevant subset. This can result in more effective data storage and utilization by end users.

Different applications can triage the data into different subsets as the notion of data relevance is intimately related to the context of the application. For example, data about a patient that is relevant for one disease may be less relevant for another disease. Adaptive triaging of data into different subsets based on the application can result in more targeted utilization of the data. If data storage constraints are paramount, only data that is relevant for the set of applications under consideration need to be stored, thus potentially reducing data storage costs.

Existing approaches to data reduction typically involve "feature reduction" where the number of features associated with the data is reduced. Such methods do not typically filter the data at the data record level but rather reduce the number of features of each data record. Providing a "data record – centric" means for data filtering can avoid utilizing data records that are noisy for subsequent analysis. For example, building a model of adverse health events can be significantly improved if less informative data records are excluded during model building. During model utilization, test data records can be similarly triaged so that less informative test records are identified as too noisy for accurate prediction rather than being used to make a possibly erroneous prediction. In health care applications for example, making erroneous predictions can be especially harmful versus flagging additional examination of an ambiguous health record.

During the period of performance for ONR Contract N00014-08-C-0036, novel computationally efficient means for performing data filtering at the data record level have

been developed. The filtered data is then used to automatically build and use improved models, and to generate and test hypotheses. In modeling complex multi-scalar systems, existing approaches model each domain with significant detail, and subsequently link the domain models into a hierarchical manner to represent the global system. However, such an approach is inefficient in dealing with complex systems with vast amounts of data. Filtering the data as described below can potentially result in simpler, more informative models of complex systems where only relevant data is used to build and test models and hypotheses.

**Data Filtering & Data Relevance:**

There has long been recognition of the need to remove irrelevant or noisy data from data sets, both in the case of data sets with defined target states as well in more general, unsupervised data sets with no target state explicitly defined. (Wilson, D. "Asymptotic properties of nearest neighbor rules using edited data", IEEE Trans. on Systems, Man and Cybernetics, 2, 408-421 (1972)). Wilson (1972) has used nearest neighbor classifiers to prefilter data for subsequent classification using a second stage classifier. In Brodley, C.E. and Friedl, M.A. "Identifying Mislabeled Training Data", J. Artificial Intelligence Research, 11, 131-167 (2005), Brodley and Friedl (2005) and references contained therein survey multiple filtering methods using ensembles of classifiers that serve as an ensemble filter for the training data. In their paper, the classification method was based on C4.5 decision trees. More generally, Brodley and Friedl describe a process whereby **m** learning algorithms are used to define an ensemble of classifiers that are then combined through a **n**-fold cross validation on the training data to filter out those data records that do not receive a requisite fraction of correct classifications. The improper classifications can be due to either a mislabeling of the target class or due to noise in the input features associated with the record of interest.

Once the first stage filtering has been accomplished, a new classifier or ensemble of classifiers can be trained on the remaining data, possibly using different classification techniques from those used during the filtering process. In the event that the target class has been mislabeled, removal of the suspect data records can improve the generalization of models trained on the properly labeled data; however, as Quinlan points out, if improper classification is due to noise in the input features associated with the training data, removing this data might not result in better models if the noise levels are high. Quinlan, J.R. "Induction of decision trees", Machine Learning, 1,81-106 (1986).

The implicit assumption here is that removal of noise during training without removing similar noise during testing may result in training models that do not reflect the noise inherent in the test set.

In the work performed under ONR Contract N00014-08-C-0036, no classifiers are used to filter data sets: A classifier makes a prediction around the target state for a given data record. In our work, the mutual information of defined ranges of one or more interacting input features against the target feature is used to identify an informative filter over a set of training data. If a new data record satisfies the rules embedded in the filter by satisfying the data ranges of the corresponding input feature combination that define the filter rules, the record is deemed to be relevant, regardless of its specific target state. In

the present approach, there is thus no explicit measurement or prediction of the target feature that is used to determine data relevance. As such, our approach is well suited to address the situation where the dominant error mechanism is inherent noise in the data environment rather than error in the labeling of the target feature. In contrast, the latter error mechanism provides the motivation and rationale for the prior art cited above.

In addition, the same filter or sets of filters that are identified on training data can further be applied against test data to remove noise in the test data prior to feeding the data into models developed using filtered training data. "Triaging" the data in this manner prior to evaluation by models can help alleviate the concern raised by Quinlan around the subsequent applicability of models trained on filtered training data to new data. In many applications, identification of relevant data prior to modeling can result in the significant reduction of both false positives and false negatives resulting from the modeling process. Instances of such error reductions will be presented below on an example data set. We note that any modeling technique that can be applied against the unfiltered data set can be applied against the filtered data set. The data filtering step has thus been decoupled from the subsequent modeling step allowing general applicability of the methods described below.

More recently, association rules analysis has been used to filter data based on informative data associations around the input features. Xiong et al (2006) have described such an approach aimed at enhancing data analysis with noise removal. Xiong, H., Pandey, G., Steinbach, M. and Kumar V., "Enhancing Data Analysis with Noise Removal", IEEE Transactions on Knowledge and Data Engineering, Vol. 18, No. 3, 304-318 (2006) and references contained therein. In such an unsupervised setting, the explicit linking to the class label (or "target state") is not established during the determination of relevance. Rather, outlier behavior of the data based solely from the standpoint of the characteristics of the inputs is what is measured as the basis for establishing relevance. Xiong et al further use association rules analysis as a means for selecting individual features for relevance rather than data records in their entirety. Their approach fits the general approach of dimensionality reduction through feature selection more than the determination of whether a data record in its entirety should be triaged.

Vaidyanathan et al in U.S. Patent 6,941,287 *Distributed Hierarchical Evolutionary Modeling and Visualization of Empirical Data*, teach methods of performing dimensionality reduction through the use of the Nishi informational metric to identify informative feature associations. They do not however teach the idea of triaging data records in their entirety to identify more relevant data subsets from a larger data environment. A key advantage of the current approach lies in the two stage process for noise filtering wherein irrelevant data records are removed in their entirety from the modeling and simulation environment and the remaining relevant data records are then further analyzed to identify the most informative feature associations. This two-stage process for noise filtering can result in models that are both more compact due to the removal of irrelevant data as well as more informative due to the identification of informative feature associations.

Thus, there is a long standing need for simplifying databases and providing a significant reduction in complexity and the resultant computational efficiency in generating models

and modeling components which results from identifying the most informative statistical relationships across large and ever increasingly complex data environments.

One advantage of the present approach is that the identification of feature filters is generally much simpler computationally than the cost of building ensembles of first stage classifiers, thus facilitating scalability. In data environments with a limited number of features (less than or on the order of 20 features), exhaustive methods can be used to measure the mutual information content of low order feature combinations from which filters can be extracted. For more complex data environments involving a larger number of features, genetic algorithms or other searching methods can be used to identify a set of informative feature combinations from which filters can be extracted. For many classification techniques, identifying informative features represents only the first step in model building. Following feature selection, further computational cost is incurred in building the model structures themselves. This cost can be alleviated using the methods of the present approach.

Another key advantage of the present approach is related to the capability of providing a new way of viewing distributed modeling. In the present approach, the feature filters span the input feature space. If there is sufficient coverage across the feature space, the resulting filtered data set can provide the basis for a robust model, even if the filtering results in a relatively small training set. In this sense, the term "distributed" refers to building a model using data that is filtered through feature filters that are distributed across the feature space. This is in contrast to the more conventional usage of the term "distributed" that involves building models that are further distributed across the data space. This has significant consequences for building scalable analytic solutions, since generally the number of features is much smaller than the number of data records. The underlying assumption of the present approach is that it is sufficient in general to build relatively few models that span the feature space using smaller amounts of data where the irrelevant data has been removed. Current state of art ensemble based modeling methods typically involve the generation of large numbers of models distributed over significantly larger fractions of the data space, and assume that the models act as data filters concurrently while making predictions. In the present approach, identifying informative feature filters that span the feature space provides a basis for first separating the removal of irrelevant noise from the subsequent step of building models. Viewing a model as a signal to noise amplifier, this amounts to increasing the signal to noise of an individual model significantly by first removing the noise from the data environment, before feeding the data into the amplifier. As a result, fewer and smaller models can be used to represent large data environments.

The informative feature filters described in the present approach can further be used to drive dynamic simulations directly from empirical data. An informative filter encodes probabilistic associations between a combination of input features and a target feature.

These probabilistic associations, learned directly from the data, can be invoked stochastically during a dynamic simulation by modeling entities such as agents in an agent based modeling environment to drive emergent behavior characteristic of complex, adaptive systems. Linking one or more filters to dynamic data sources that are derived from either real or synthetic data, can additionally be used to drive simulations using

updated data inputs. Therefore, in addition to using feature filters to prefilter data prior to the automatic generation of signal rich models, the filters can be used directly to drive dynamic simulations of complex, adaptive systems.

The present approach further describes methods for constructing optimum combinations of filters to identify relevant data. The methods of the present approach allow optimum filter combinations to be represented as a composite database query. The resulting query can then be resolved by the query processing engine resident within the database to retrieve informative data to either the end user or for other analysis applications. The retrieved data is information rich against a user specified target feature, enabling the user to gain an "informative view" (or *Info View*) of the underlying database. This capability can significantly enhance the value of the database to the end user by isolating relevant data embedded within increasingly larger database environments. We note that the methods of the present approach can be applied across multiple databases with the info views from each database aggregated to present a composite view to the end user or application.

Finally, the present approach addresses the issue of filtering entire data records from further analysis. This is distinct from the well studied problem of feature selection in machine learning described for example by Bishop and in references contained therein where the goal is to reduce the dimensionality of a data set prior to modeling. Bishop, C.M., "Neural Networks for Pattern Recognition", Oxford University Press, USA; 1 edition (1996) and references contained therein. In such a case, all the data records are maintained, but "irrelevant" features are removed across all the records. The present approach supports the application of feature selection methods on a data set which has been pre-filtered at the data record level in order to create the most "signal rich" data environment for modeling and analysis.

In summary, we present a new approach to the removal of irrelevant data. The fundamental idea is based on the identification of informative "feature filters" that represent combinations of input features that preferentially filter data with respect to a specific target. Mutual information metrics are used to measure the information content of a feature filter with respect to a target feature. The feature filters inherently encode informative interactions between features through the inclusion of explicit ranges of values for each feature in multiple feature combinations that are evaluated concurrently. The present approach includes methods for automatically identifying multiple feature filters that exceed a mutual information threshold. The selected feature filters are then aggregated to form a composite filter set that is used to remove irrelevant data. The present approach further defines methods for identifying optimal values for the mutual information threshold to determine the optimum composite filter. For emphasis, we note again that no explicit classification of an individual data record with respect to a target state is performed during the filtering process. Rather, a data record is deemed to be irrelevant if its feature characteristics do no match those in the aggregated set of feature filters. The role of the target feature is therefore encoded in the information content of the filter, not in the specific target state of an individual data record.

**Identification of relevant data - Details:**

The methods of the present approach offer unique capabilities in identifying relevant subsets of data that may be embedded in large data environments. Based on the principle of building data management and analysis capabilities in a modular, progressive fashion, subsets of data that result from relatively simple informative and relevant "clusters" that are automatically identified are combined in several ways to provide the basis for subsequent modeling and analysis as well as to obtain insight. Individual data clusters can be combined optimally via both *union and intersection* operations using optimization techniques. An optimal union of clusters can facilitate the generation of larger, "relevant" clusters that are informative and less noisy for subsequent model building (Figure 1). An optimal intersection of clusters can reveal more specific sub-clusters that can isolate and present interesting subsets of data to the user for analysis and understanding (Figure 2).

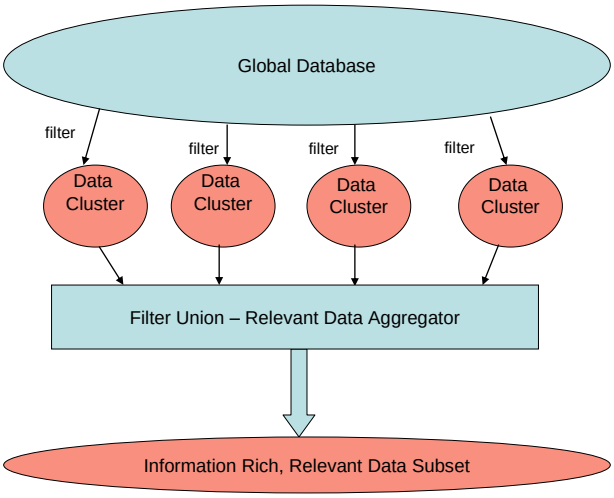


*Figure 1. The aggregation of multiple signal rich local data clusters to form a larger relevant data subset*

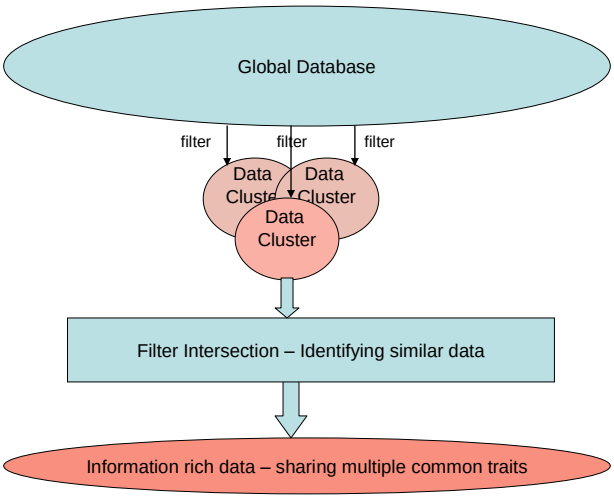


*Figure 2. The intersection of multiple signal rich data clusters to identify an informative data subset that shares multiple common traits*

It should be noted that relevance is measured with respect to a specific target or question. A particular data set can have high relevance to one target but low relevance to another. In the method of the present approach, informational metrics are used to measure the relevance of a data set to a target, and automated methods (through the union and intersection operations mentioned above) have been developed to generate high relevance data subsets from larger data sets.

*Identification of an optimal union of data clusters:*

An optimal union of multiple signal rich data clusters is identified using the following methodology:

a. An interval of mutual information thresholds for data clusters ranging from a minimum mutual information threshold to a maximum mutual information threshold is defined. Note that each cluster is derived from a corresponding "data filter" that represents a combination of input features where each feature is in a specific state.

b. For each mutual information threshold, a set of data filters is automatically identified where the mutual information of the underlying data cluster exceeds the threshold, and where the data support for the cluster exceeds a minimum data

support level. The filters can be identified either by exhaustive searching or by other searching techniques such as genetic algorithms.

c. An aggregate data set resulting from the merging of all the data clusters from step (b) is then assessed for mutual information against the target feature, using the mutual information metric:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left( \frac{p(x,y)}{p_1(x)\, p_2(y)} \right),$$

where $p(x,y)$ is the joint probability distribution function of $X$ and $Y$, and $p_1(x)$ and $p_2(y)$ are the marginal probability distribution functions of $X$ and $Y$ respectively. Here, X represents an input feature, and Y represents the target feature. Note that the merging of the individual data clusters can also be expressed in terms of the union of the corresponding data filters.

d. As the mutual information threshold is increased from its minimum value, the mutual information profile for each corresponding aggregate data set is analyzed to identify the threshold value where there is both a sharp increase in the mutual information of the aggregate data as well as a sharp decrease in the level of data support. The degree of sharpness in the discontinuity is controlled by the user. The filter union and corresponding data aggregate at this point of discontinuity defines the "signal rich" data useful for further study.

*Identification of an optimal intersection of data clusters:*

An optimal intersection of multiple signal rich data clusters is identified using the following methodology:

d. A set of information rich input feature combinations against a target feature is automatically identified from the data. This identification can be enabled by either exhaustively searching the input feature space or by using other searching techniques such as genetic algorithms. Note that each selected feature combination consists of multiple data filters where each filter represents a unique set of feature states associated with the combination.

e. Defining a fitness function that comprises both a data support term and a feature complexity term across one or more intersecting data filters:

fitness function = $\lambda$*data support – ( 1- $\lambda$)/(feature complexity)

where $\lambda$ is a normalized tuning parameter between 0 and 1 that adjusts the relative weighting of data support versus feature complexity.

f. Searching the space of informative data filters across each feature combination in step (a) for a combination of intersecting data filters that maximizes the fitness function of step (b).

For example, if $\lambda$ is set to 1, data support becomes the dominant factor controlling fitness, and a single filter that provides maximum data support will be selected. Conversely, if $\lambda$ is set to 0, feature complexity as defined by the number of features participating in the intersecting filter set becomes the dominant factor. In this instance, a maximal number of filters will be selected, regardless of the resulting data support. For intermediate values of $\lambda$, a pool of "hybrid" filter intersections can be identified that balance the weighting of data support with that of feature complexity. The end result is a set of intersecting data records that share multiple common feature states.

The underlying premise around data relevance is that more informative "signal" models can be built from high relevance data sets. In effect, much of the noise in the data has been filtered out, leaving an information rich data "kernel" that can be explored and modeled. New test data coming in can be assessed by the relevance filter with the data that passes the relevance test representing signal that can effectively be modeled. Thus, noise can be filtered out of the system both during model building as well as model usage. The ability to automatically separate data that represents "signal" from data that represents "noise" during both model building and model usage is an important differentiating capability of the present approach. Typically, this separation does not occur in data management/analysis systems, or the separation is based on a predefined noise model that is imposed on the data. The ability to automatically separate out noise data from signal data can have important consequences in subsequent decision making; for example, ignoring predictions from irrelevant data and only acting upon predictions from relevant data can improve the overall effectiveness of decision making.

The capability of automatically aggregating relevant data across one or more databases to provide an informational view (*Info View)* into the data environment is an important differentiating capability of the present approach. Traditional data views within a database environment result from associations made only at the data level. Using informational metrics to guide the automatic generation of informative data views that can be processed by both human end users as well as other analytic/data processing tools provides a basis for transforming data warehouses into information warehouses. This capability has significant implications in driving an effective and scalable transition from data to information to knowledge. Analysis engines can use less data that is more relevant to the target at hand to build more accurate signal models that can be used to generate and test hypotheses, make predictions and gain insight. In a data environment that is continuing to expand rapidly, this capability will become increasingly important.

The intersection of data records over multiple data clusters represents a powerful way to present interesting data to the user to gain insight as well as facilitate hypothesis generation. Data that share multiple common feature traits, extracted from a much larger database, can provide insight into interactions that are informative against a particular target. The methods of the present approach automatically generate such interesting data to the end user and/or other analysis and visualization applications.

An interesting example of the identification of intersecting data records within a large database presents itself in the area of combinatorial chemistry. Chemical compounds are often described by the presence or absence of chemical substructures. Discovering

compounds that share multiple structural features that map to biochemical activity can provide a useful guide to elucidation of activity mechanisms as well as guide synthetic drug design. In addition, using the intersection of data records over multiple low dimensional data clusters to identify high dimensional commonalities can be significantly more efficient than directly searching across a high dimensional space.

Note: An end user can drive the automatic generation of composite filter query to retrieve data that is relevant against a user defined target. The retrieved data can be used by both the end user and/or analytic tools for hypothesis generation and model building.

Figure 3 outlines the coupling of a relevance filter into a database environment to provide "Info-Views" around data relevant to a specific target or set of targets. An end user can define a target (or targets) of interest and the methods of the present approach can be used to automatically generate a composite filter query to drive the retrieval of relevant data into an "Info-View". We note that both the union and intersection operations that are applied to the database can be expressed in the language of database filtering. The union operation represents a logical OR-ing of several individual filters that define the informational clusters and the intersection operation represents a logical AND-ing of several individual filters. Thus, existing methods for resolving database queries can be applied seamlessly to the relevance filter of the present approach in order to present informative data views to the end user or analysis application. This helps address some important issues around scalability, as the relevance filter can be implemented as a thin layer on top of existing database systems and leverage already existing and optimized methods for generating data views in large data environments. Distributing the filtering capability across multiple data subsets spanning the database can further improve scalability by generating multiple, smaller informative data views that could provide the basis for distributed modeling. Finally, we note that the database environment could represent more than one database as the process outlined above could be executed simultaneously across multiple databases, with each separate Info-View being merged into a final composite Info-View.

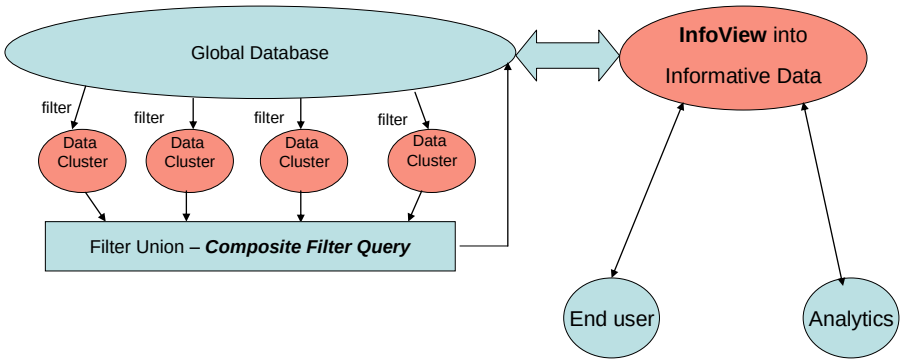


*Figure 3. Providing "InfoViews" into data base environments. An end user can drive the automatic generation of composite filter query to retrieve data that is relevant against a user defined target. The retrieved data can be used by both the end user and/or analytic tools for hypothesis generation and model building.*

**Automatic Building of Signal Models from Relevant Data Subset:**

The methods of the present approach also provide for the capability of automatically generating one or more *signal* models from informative data subsets for predictive analytics and hypothesis generation/testing. It should be noted that *any* empirical modeling technique that can model a global data set can also be used to model an informative data subset that has been automatically identified from the global data. Examples of modeling techniques include decision trees, neural networks, Bayesian network modeling, and a variety of both linear and non-linear regression techniques. Using the methods of the present approach to first identify relevant data subsets from which populations of models are then automatically generated, can result in improved signal models that are modeling the information embedded in the data rather than the noise. Traditional modeling paradigms generally do not automatically separate signal from noise at the data record level during the process of building models; rather, variables are preferentially selected that tend to be more informative across the *entire* data set. Feature selection that occurs as part of model building is thus a primary means for noise removal in current modeling approaches. In the methods of the present approach, there is both data record filtering as well as feature filtering to reduce the noise in the data environment for a particular modeling application. The data record filtering using automatically generated relevance filters presents a key differentiator between the current approach and other data management/analysis systems.

Note: First, the number of records is reduced, followed by feature filtering on the reduced database.

Figures 4 and 5 compare traditional noise filtering against noise filtering as described in the present approach. In Figure 4, the number of columns, or features, is reduced during the feature selection sub step of model building. Note that the number of rows, or data records, is preserved during feature selection. In Figure 5, the first step involves reducing the number of data records by removing irrelevant records that do not satisfy the rules described by the composite filter union. Traditional feature selection methods can then be applied as a second step on the reduced data set. The application of both noise reduction steps in the present approach can result in the generation of superior hypotheses and predictive models as will be demonstrated in the example below.

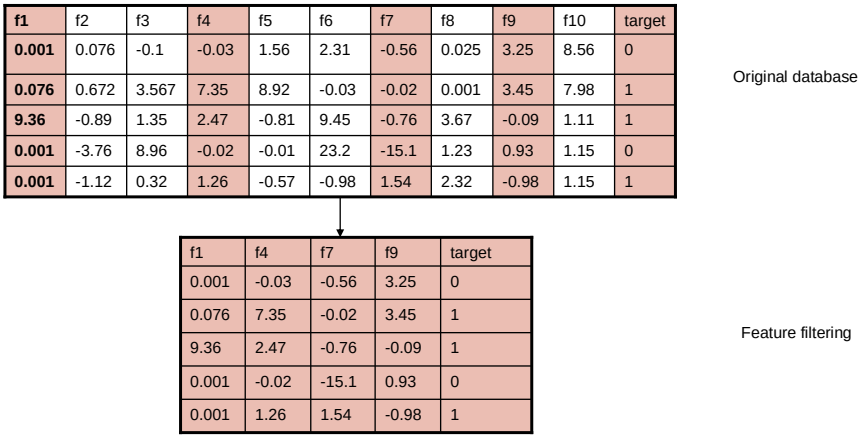


*Figure 4. Traditional feature selection approach to noise reduction. Note that the number of rows remains unchanged during feature selection.*

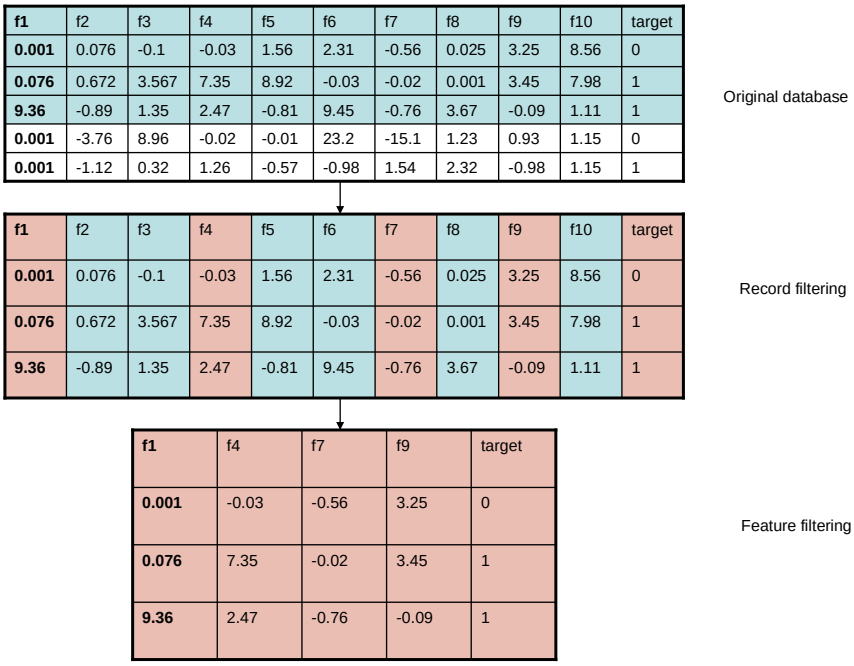


*Figure 5. Noise filtering approach in present invention. First, the number of records is reduced, followed by feature filtering on the reduced data base.*

*Using Informative Filters to drive dynamic simulations:*

The informative filters and filter combinations described in the present approach can be used to define informative rules that can drive dynamic simulations. Agent based modeling is a modeling paradigm that is particularly well suited to this approach, where the behavior of individual agents, representing modeling entities, can be driven stochastically by the probabilistic rules embedded in the filters associated with the agents. Such a modeling paradigm, driven by rules that are learned directly from the data, can result in emergent behavior of the global modeling environment that is well matched to observations.

Informative Filters can also be used to identify a group of modeling components that are mutually informative or that together are informative against a specific target or targets. Identifying subsets of "signal rich and noise poor" informative modeling components within a large data environment can reduce the complexity of subsequent models and simulations without suffering a significant loss in modeling fidelity.

Alternatively, the simulations can generate new data during a simulation run that can in turn be assessed by the filters to modify the subsequent dynamics of the simulation. If the simulation is coupled to an external dynamic data source, changes in the external data can further modify simulation dynamics.

**Summary:**

For completeness, key differentiators between the methods described in the present approach and prior art include:

- Automatic identification of informative and relevant data subsets using mutual information measures for subsequent model building and system understanding. This is enabled through the discovery of multiple informative clusters that are then combined through either *union* or *intersection* operations.
- Leveraging the identification of relevant data subsets into a mechanism for providing Info Views into large databases above and beyond more traditional data views. This capability, implemented through existing database filtering operations, can transform data warehouses into information warehouses. We note that the larger database could represent a virtual database comprised of one or more distinct databases.
- The ability to develop more *accurate signal* models by modeling on less noisy, relevant data subsets rather than the entire data space. Related to this is the ability to automatically separate signal from noise during model building and model usage through both feature filtering as well as data record filtering. Again, we emphasize that different existing modeling paradigms can be used to generate the signal models on the relevant data.
- The capability for developing more *scalable* analytics by modeling on relevant data subsets rather than the entire data space.
- The ability to use the probabilistic rules embedded in the filters, learned directly from the data, to drive dynamic simulations.

## Work Flow of LeapWorks Data Analytics:

This section walks the user interactively through the major steps of the LeapWorks Pattern Discovery and Predictive Analytics tools through a succession of screenshots. In order to provide a context for this exercise, the following motivating example is used:

### Motivating Example: KDD Cup 2008 - Early detection of breast cancer

*(Note: The Background and Data Descriptions have been adapted from the information provided in http://www.kddcup2008.com/KDDsite/KDDcup2008.htm).*

#### *Background*

*Breast cancer* is a disease in which malignant (cancer) cells form in the tissues of the breast. Breast cancer is the second leading cause of cancer deaths in women today (after lung cancer) and is the most common cancer among women, except for skin cancers. About 1.3 million women are expected to be diagnosed annually with breast cancer worldwide, and about 465,000 will die from the disease. In the United States alone, in 2007 an estimated 240,510 women were expected to be diagnosed with breast cancer, and 40,460 women are expected to have died from breast cancer.

*Screening* is looking for cancer in asymptomatic people – i.e., before a person has any symptoms of the disease. Cancer screening can help find cancer at an early stage. When

abnormal tissue or cancer is found early, it is often easier to treat. By the time symptoms appear, cancer may have begun to spread. The good news is that breast cancer death rates have been dropping steadily since 1990, both because of earlier detection via screening and better treatments.

The most common breast cancer screening test is a *mammogram*. A mammogram is an x-ray of the breast. The ability of a mammogram to find breast cancer may depend on the size of the tumor, the density of the breast tissue, and the skill of the radiologist. The mammogram is considered the standard of care for most asymptomatic women. For instance, in the US , insurance companies routinely reimburse for an annual screening mammography examination, for all asymptomatic women over the age of 40. These exams are credited with reducing the breast cancer death rate by approximately 30% since 1990.

However, the reading of screening mammograms is challenging. Findings on a screening mammogram leading to further recall are identified in approximately 5%-10% of patients, even though breast cancer is ultimately confirmed in only three to ten cases in every 1,000 women screened. Perhaps even more importantly, there is compelling evidence that many breast cancers detected at screening mammography are, in retrospect, visible on the previously obtained mammograms but have been missed by the interpreting radiologist in the prior year. There are several reasons for this: The complex radiographic structure of breast tissue, particularly in dense breasts; the subtle nature of many mammographic characteristics of early breast cancer; human oversight; poor quality films and even fatigue or distraction are all reasons why cancer is not detected by mammography.

To overcome the known limitations of human observers, second (ie double) reading of screening mammograms by another radiologist has been implemented at many sites. Studies indicate a potential 4%-15% increase in the number of cancers detected with double reading. In a radiology practice that performs 10,000 screening examinations per year, generally between 30-100 cancers per year will be detected. Thus, double reading in this practice could contribute to the diagnosis of 1-15 additional cancers per year. However, this approach results in a doubling of the radiologist-effort so it is not financially viable.

Rapid and continuing advances in computer technology, as well as the ready adaptation of radiology images to digital formats, have increased the interest in computer prompting to enable the attending radiologist to act as his or her own second reader. One very promising adaptation of computer-prompting technology is computer-aided detection (CAD) in screening mammography. Current CAD systems demonstrate a high rate of detecting cancerous features on mammograms, but further improvements in both sensitivity and specificity would lead to tremendous benefits both in terms of lives saved each year, and in terms of reduction n the workload of radiologists. For the last 8-10 years, US insurance companies have begun to provide additional reimbursement to mammographers who run CAD algorithms on the mammograms – in other words, physicians are now reimbursed for running a machine learning algorithm to help them better detect cancer.

In an almost universal paradigm, the CAD problem is addressed by a 4 stage system:

1. candidate generation which identifies suspicious unhealthy candidate regions of interest (candidate ROIs, or simply candidates) from a medical image;

2. feature extraction which computes descriptive features for each candidate so that each candidate is represented by a vector x of numerical values or attributes;

3. classification which differentiates candidates that are malignant cancers from the rest of the candidates based on x; and

4. visual presentation of CAD findings to the radiologist.

***Data***

A breast cancer screen typically consists of 4 X-ray images; 2 images of each breast from different directions (these views are called MLO and CC). Thus, most (but not all) patients would have MLO and CC images of both their breasts, giving a total of 4 images per patient. Each image is represented by several candidates. For each candidate, we provide the image ID and the patient ID, (x,y) location, several features, and a class label indicating whether or not it is malignant. We provide features computed from several standard image processing algorithms – 117 in all. ***(Note:*** In the actual data set, the feature labels were anonym zed. For ease of interpretation for our walk through example, we have labeled the features with labels that are used commonly for this type of image analysis. We emphasize that these labels are only representative and do not represent ground truth).

The target labels indicate whether a candidate is malignant or benign (based on either a radiologist's interpretation or a biopsy or both). Note that several candidates can correspond to the same lesion. Thus, we also provide a unique lesion-ID for the malignant lesions in the data.

Information is provided for a set of 118 malignant patients (patients with at least one malignant mass lesion). We also include data from 1594 normal patients – where all candidates are presumed to be benign. The training set consists of a total of 50563 candidate ROIs, each described by 117 features, but only an extremely small fraction of these 102,294 candidates is actually malignant. The test set consists of a total of 51731 candidate ROIs.

***Challenge***

The rate of prevalence of malignant patients in a screening environment is extremely low (on average only around 5-10 patients out of 1000 screening patients have breast cancer). The challenge is to discover informative patterns in this "needle in a haystack" type of data as a basis for building predictive models from the training data to accurately identify malignant ROI's in an out of sample test set.

***PATTERN DISCOVERY WORK FLOW:***

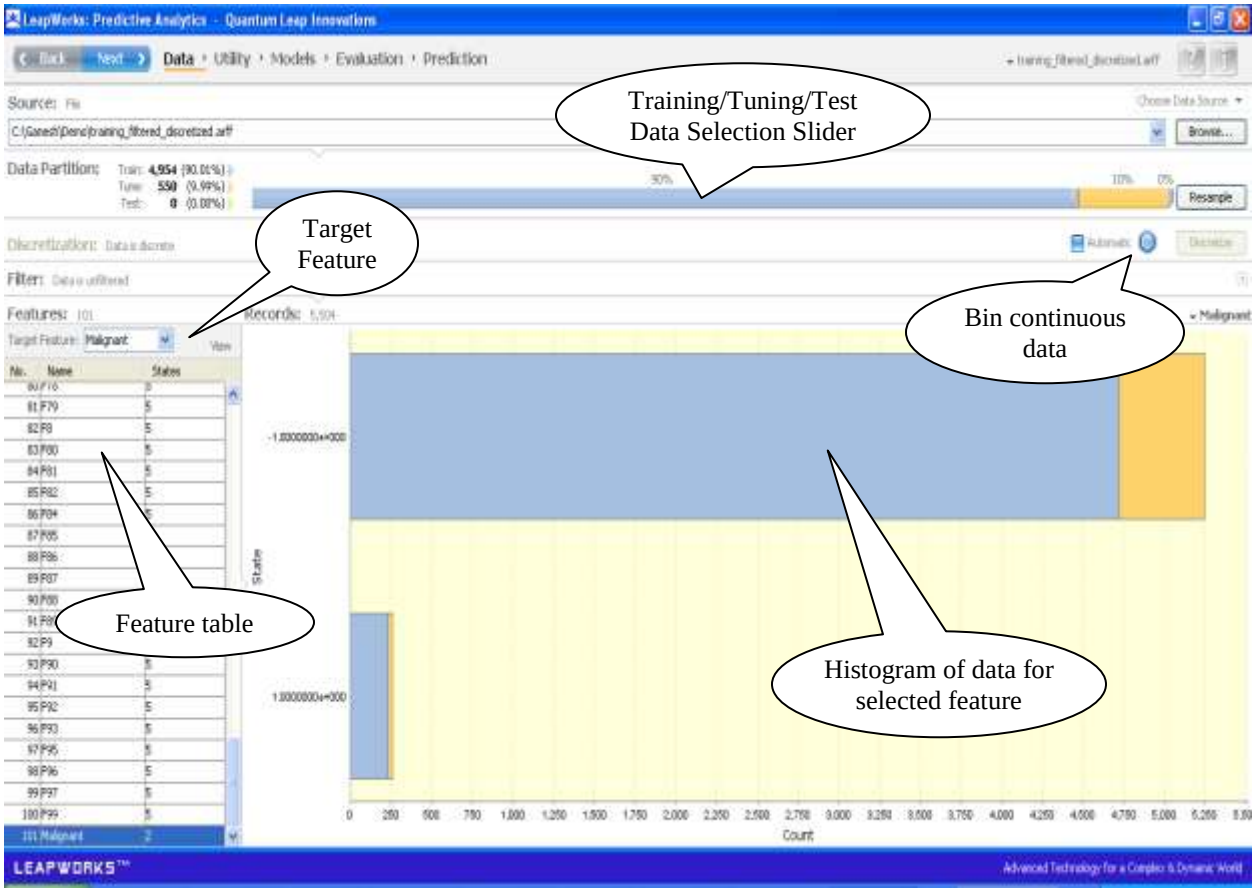


*Figure 6. Data Screen.*

**Data Screen:**

- The Data Screen enables data loading from a variety of data sources, including files and relational databases.
- The user can partition the data into training/tuning/test data subsets using a slider.
- In addition, there are mechanisms for preprocessing the data prior to entry within the LeapWorks environment.
- Continuous data can be binned into discrete states by pressing the "Discretize" button.
- The user can select the target feature for analysis.
- The user can highlight a data feature from the Feature Table and view the corresponding data distribution.


*Figure 7. Patterns Screen.*

**Patterns Screen:**
The Patterns Screen shows three plots:

- The plot on the left shows (up to) the 100 most informative individual features against the target feature. The slider directly below the plot controls the number of features that are used to identify the most informative multi-dimensional feature combinations. The slider below this slider controls the makeup of the selected features – e.g., the most informative features versus randomly selected features.
- The middle plot shows the most informative multi-dimensional feature combinations against the target feature. The number of feature combinations can be selected using the slider in the bottom right. The dimensionality (or "complexity") of the feature combination can be controlled using the slider in the bottom middle.
- The plot on the right shows a frequency distribution of how often an individual features is present in the most informative multi-dimensional feature combinations. Highlighting a selected feature in this plot highlights the corresponding multi-dimensional feature combinations in the middle plot.

*Notes:*
- Mousing over any bar in the plots will identify the corresponding feature or feature combination.
- Clicking on a bar in the middle plot will launch the Pattern Analysis and Visualization screens described below.


*Figure 8. Pattern Analysis.*

**Pattern Analysis:**

- When a feature combination is selected from the Patterns screen, the Pattern Analysis screen provides more detailed elaboration of the corresponding patterns. A pattern is defined as a specific set of (feature, state) vectors.
- The plot on this screen plots all patterns for the selected feature combination against data support and mutual information. Data support refers to the percentage of records in the entire data set contained in a pattern, and mutual information is a statistical measure of how much information against the target feature is contained within the pattern. A robust, informative pattern would be represented by a circle on the top right of the plot.
- The table at the bottom of the screen displays statistics of a selected pattern that is highlighted within the plot.


*Figure 9. Pattern Visualization.*

**Pattern Visualization:**

- When a feature combination is selected from the Patterns screen, the Pattern Visualization screen provides more detailed visualization of the corresponding patterns. A pattern is defined as a specific set of (feature, state) vectors.
- The plot on this screen displays all the patterns associated with a feature combination using views that can be selected by the user. Patterns can be visualized based on data support, amount of information and the dominant target feature state associated with the pattern.
- The table at the bottom of the screen displays statistics of a selected pattern that is highlighted within the visualization table.

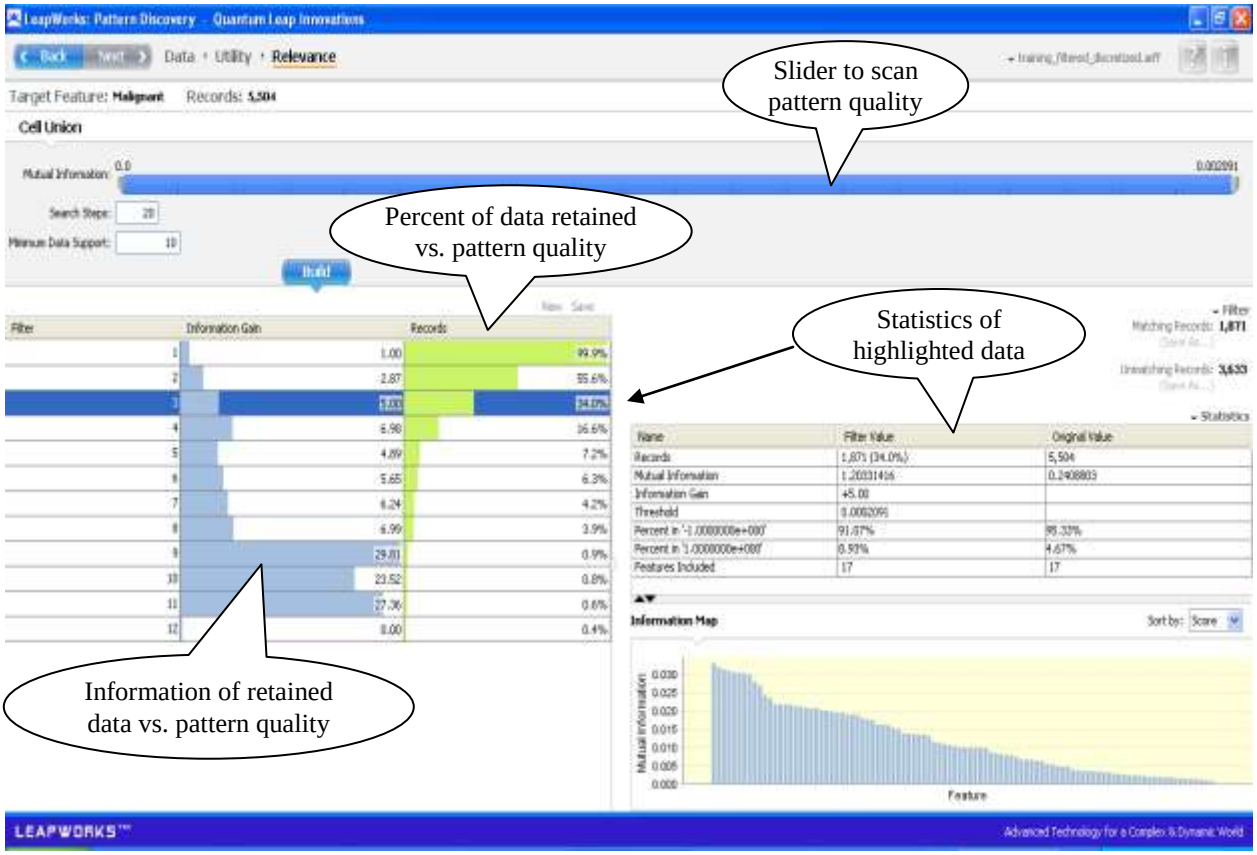


*Figure 10. Filtering.*

**Filtering:**

- The Filtering Screen aggregates multiple patterns to create a composite pattern that acts as a data filter to both reduce the amount of data as well as enrich the information contained within the data.
- Composite patterns can be built using different thresholds for the "quality" (as defined by information strength) of the constituent patterns. This results in the two colored profiles displayed in the lower left of the screen.
- When a particular composite pattern is selected, the statistics of the filtered data corresponding to the composite pattern are displayed in the table on the top right.

## *PREDICTIVE ANALYTICS WORKFLOW*:

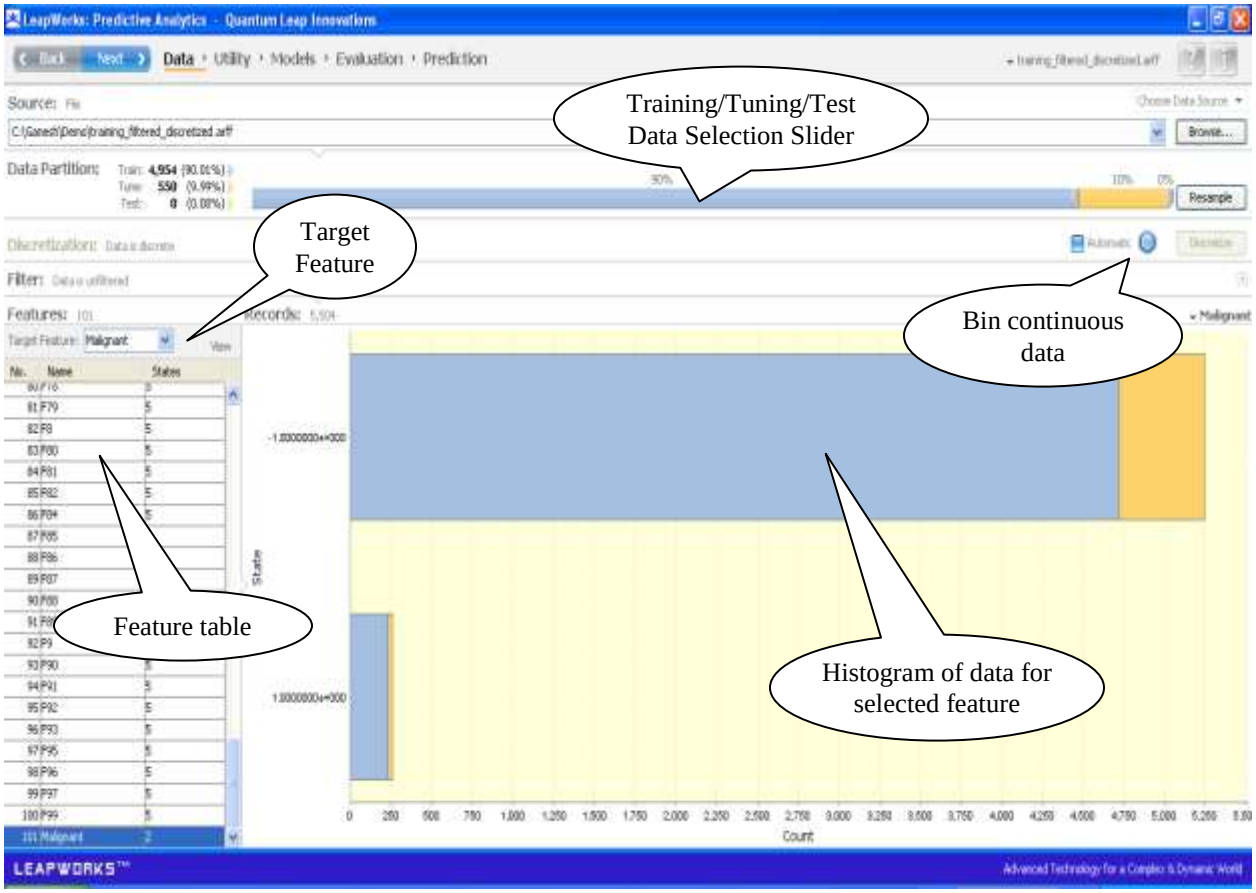


*Figure 11. Data Screen.*

**Data Screen:**

- The Data Screen enables data loading from a variety of data sources, including files and relational databases.
- The user can partition the data into training/tuning/test data subsets using a slider.
- In addition, there are mechanisms for preprocessing the data prior to entry within the LeapWorks environment.
- Continuous data can be binned into discrete states by pressing the "Discretize" button.
- The user can select the target feature for analysis.
- The user can highlight a data feature from the Feature Table and view the corresponding data distribution.


*Figure 12. Patterns Screen.*

**Patterns Screen:**
The Patterns Screen shows three plots:

- The plot on the left shows (up to) the 100 most informative individual features against the target feature. The slider directly below the plot controls the number of features that are used to identify the most informative multi-dimensional feature combinations. The slider below this slider controls the makeup of the selected features – e.g., the most informative features versus randomly selected features.
- The middle plot shows the most informative multi-dimensional feature combinations against the target feature. The number of feature combinations can be selected using the slider in the bottom right. The dimensionality (or "complexity") of the feature combination can be controlled using the slider in the bottom middle.
- The plot on the right shows a frequency distribution of how often an individual features is present in the most informative multi-dimensional feature combinations. Highlighting a selected feature in this plot highlights the corresponding multi-dimensional feature combinations in the middle plot.

*Notes:*

- Mousing over any bar in the plots will identify the corresponding feature or feature combination.
- Clicking on a bar in the middle plot will launch the Pattern Analysis and Visualization screens described in the description of the Pattern Discovery tool.

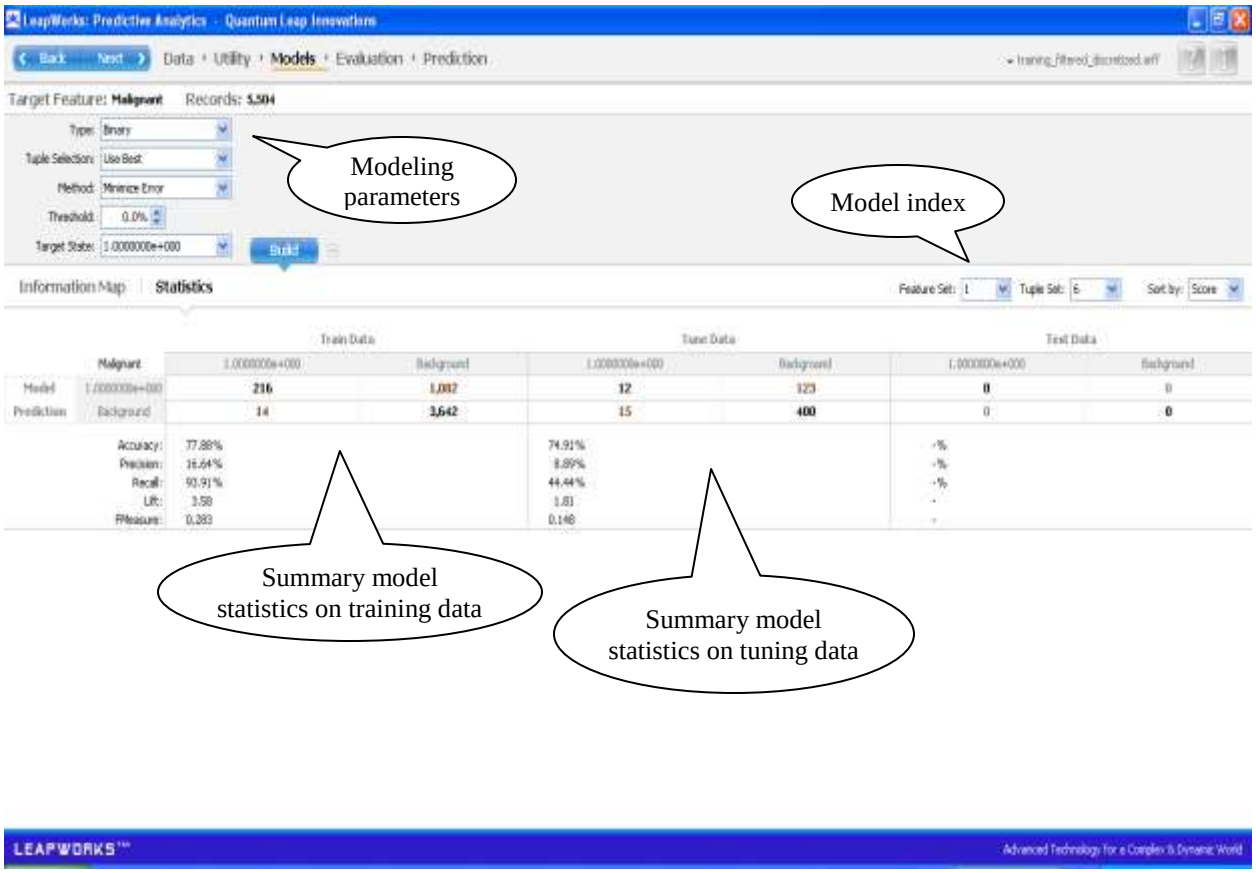


*Figure 13. Models Screen.*

**Models Screen:**

- The Models Screen enables model building using patterns that have been discovered from the Patterns screen.
- The user can select modeling parameters to drive the model building.
- The screen can display either summary statistics for the training/tuning/testing data subsets as shown or visualize the patterns that make up the model(s) using the "Information Map" tab.

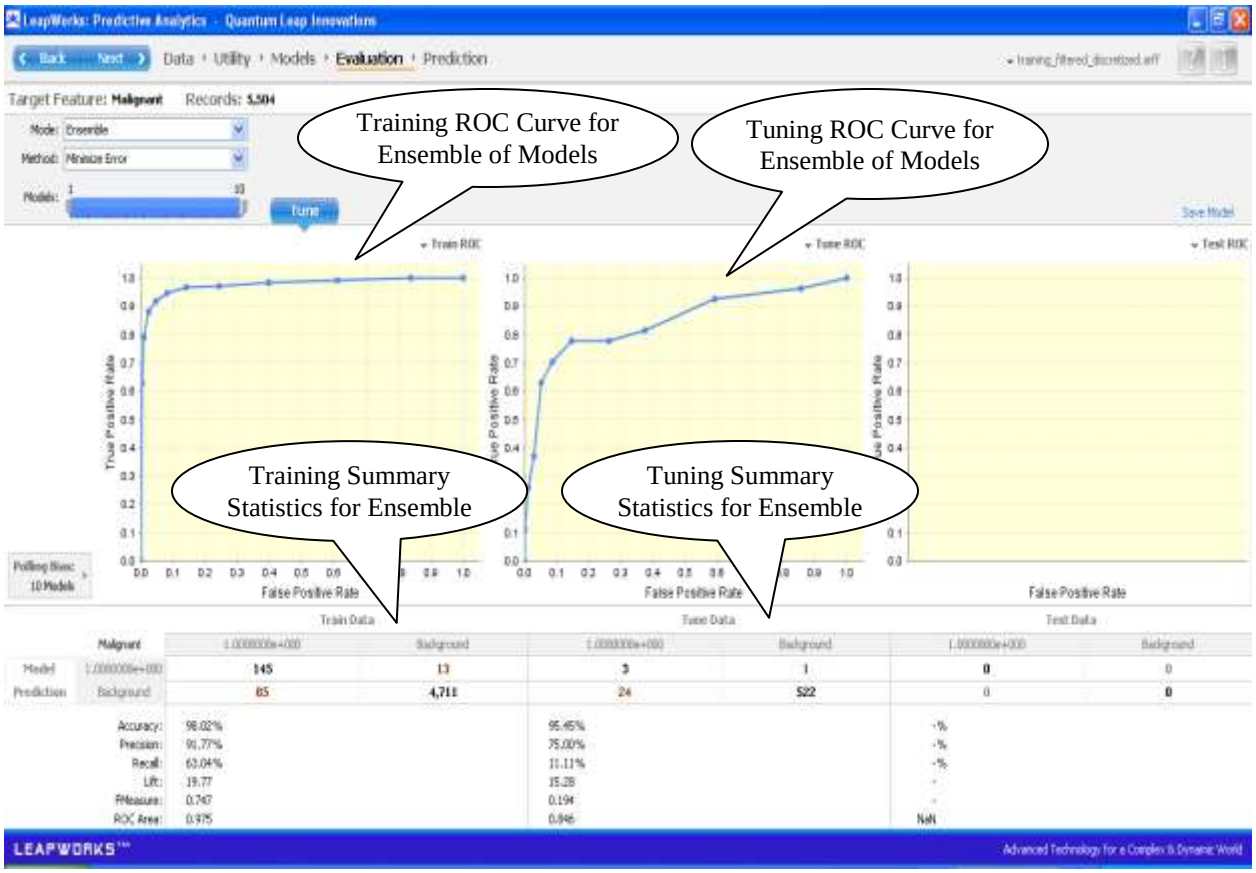


*Figure 14. Tuning Screen.*

**Tuning Screen:**

- The Tuning Screen enables model tuning using the ensemble of models that have been built from the Models screen.
- The user can select tuning parameters to drive the tuning.
- The screen displays both performance curves and summary statistics for the training/tuning/testing data subsets.

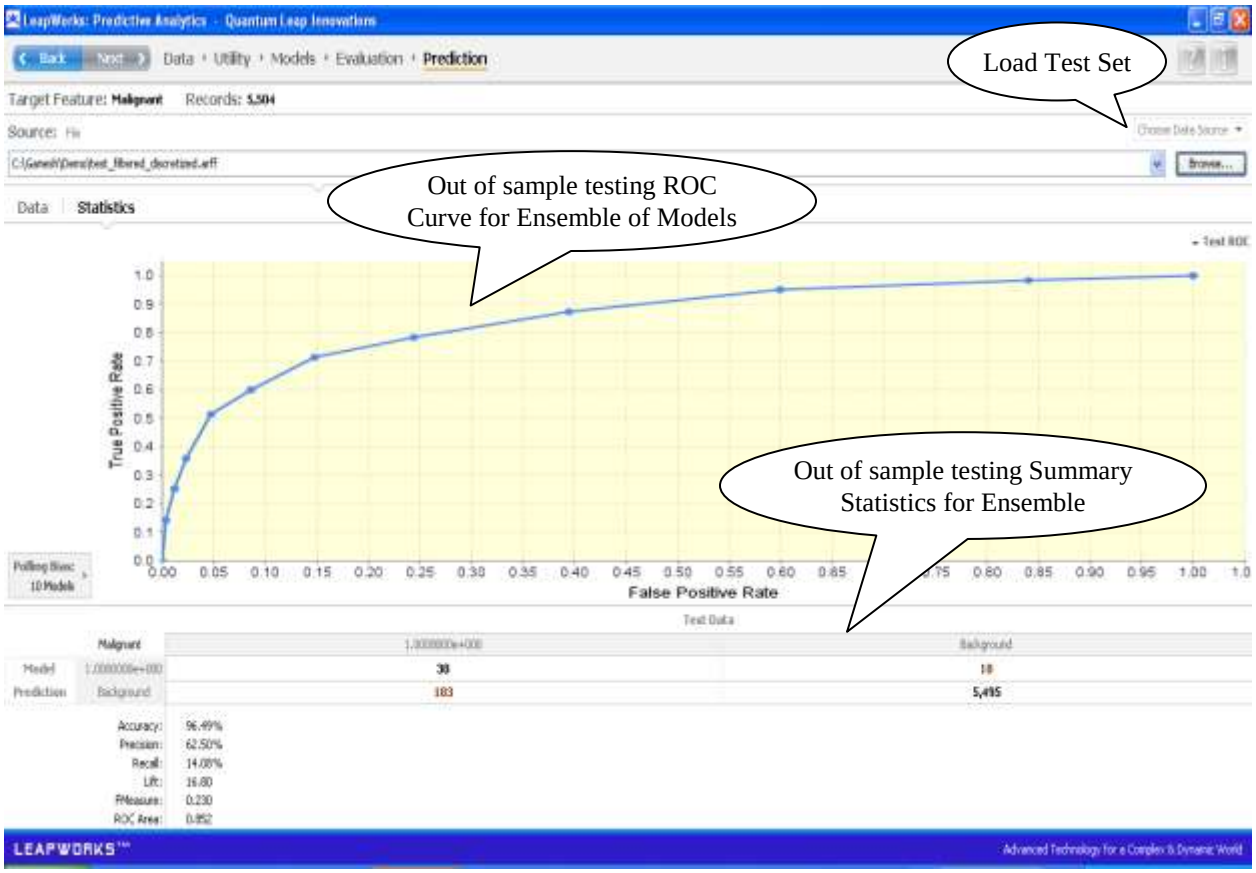


*Figure 15. Prediction Screen.*

**Prediction Screen:**

- The Prediction Screen allows the user to test the composite model that has been tuned from the Tuning screen on an out of sample test data set.
- The screen displays both performance curves and summary statistics for the out of sample test data set.

## Comparative Studies: LeapWorks Predictive Analytics versus Weka:

### Motivation:

In this summary, we present results from a comparative study of LeapWorks™ Predictive Analytics (PA) versus several machine learning techniques implemented in the Weka Open Source Machine Learning Repository. The study was performed across several data sets that were chosen both as representatives of important classes of problems as well as for the different types of challenges that they presented. We note that in a comparative study involving multiple analysis methods, it was not feasible to optimize each method to produce optimal results as the basis for comparison. Instead, default values were used for setting parameter values across all methods, including LeapWorks PA. This approach in turn influenced the selection of appropriate metrics for comparison.

In addition to "quality of results" metrics, the study also compared the performance times across the various methods. Model building time was measured for each method across the data sets as a measure of scalability. Methods were assessed in a two dimensional "quality of result versus model building time" space to assess the resulting tradeoffs.

The core comparison strategy involved the use of the original (unfiltered) data sets for both training and testing the models. In order to provide a broader assessment of the impact of the LeapWorks Data Utility and Relevance (DUR) component on subsequent analysis, four classes of experiments were performed on each data set:

1. Unfiltered training data/Unfiltered test data
2. Unfiltered training data/DUR filtered test data
3. DUR filtered training data/Unfiltered test data
4. DUR filtered training data/DUR filtered test data

Results will be provided for all classes of experiments. The summary will follow the following outline:

A. Description of Data Sets
B. Summary of Analysis Methods
C. Definitions of Metrics for comparison
D. Weka/LeapWorks PA model building protocol
E. Summary of Results
F. Discussion

## A. Description of Data Sets:

### *1. KDD Cup 1999 - Intrusion Detector Learning*

Software to detect network intrusions protects a computer network from unauthorized users, including perhaps insiders. The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between ``bad'' connections, called intrusions or attacks, and ``good'' normal connections.

The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate research in intrusion detection. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided. The 1999 KDD intrusion detection contest uses a version of this dataset.

Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. They operated the LAN as if it were a true Air Force environment, but peppered it with multiple attacks.

The raw training data was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records.

A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes.

Attacks fall into four main categories:

- DOS: denial-of-service, e.g. syn flood;
- R2L: unauthorized access from a remote machine, e.g. guessing password;
- U2R: unauthorized access to local superuser (root) privileges, e.g., various ``buffer overflow'' attacks;
- Probing: surveillance and other probing, e.g., port scanning.

It is important to note that the test data is not from the same probability distribution as the training data, and it includes specific attack types not in the training data. This makes the task more realistic. Some intrusion experts believe that most novel attacks are variants of known attacks and the "signature" of known attacks can be sufficient to catch novel variants. The datasets contain a total of 24 training attack types, with an additional 14 types in the test data only. We mapped these attack types into the four main categories identified above.

For our comparative analysis, we reduced the number of attack states to 2 ("Attack" versus "Normal") where we did not differentiate between the types of attack, but rather detected the presence of any attack type. There are 41 input features in the data. The training data consists of 370,244 records with an Attack:Normal target ratio of 4:1 making this the antithesis of a "needle in a haystack" type problem. The test set consists of 123,777 records.

***Note:*** LeapWorks PA is currently configured to model systems where there is a specific target state of interest. In a multi target state system, all target states other than the state of interest are treated as background. In the Intrusion data set, there are 4 attack types that we are treating equivalently as a generalized Attack state. In such a case, misclassification across different attack types is ignored since PA only differentiates between attack versus non-attack rather than the specific type of attack. LeapWorks PA is currently being extended to handle true multi-target state classification. We note however that the LeapWorks Data Utility and Relevance component does properly identify cells where the target variable has multiple states. It is the Predictive Analytics component that still needs to be properly generalized.

### *2. KDD Cup 2008 - Breast Cancer Identification*

A breast cancer screen typically consists of 4 X-ray images; 2 images of each breast from different directions (these views are called MLO and CC). Thus, most (but not all) patients would have MLO and CC images of both their breasts, giving a total of 4 images per patient. For the purposes of the KDD Cup, each image is represented by several candidates (see stage 1 above). For each candidate, we provide the image ID and the patient ID, (x,y) location, several features, and a class label indicating whether or not it is malignant. We provide features computed from several standard image processing algorithms – 117 in all – but due to confidentiality reasons we are unable to provide some additional proprietary features. The labels indicate whether a candidate is malignant or benign (based on either a radiologist's interpretation or a biopsy or both). Note that several candidates can correspond to the same lesion.

The training set consists of 50,563 data records with 118 features including all 4 views for each patient plus the target feature. The proportion of Malignant tumors to Benign Tumors is 1:151, making this a "needle in a haystack" type problem. The test set consists of 51,731 records that were sampled from the original KDD Training Data since target states were not provided for the original KDD Test Data.

### *3. Predicting Plays in the NFL:*

The objective is to predict whether the opposing team will run or pass at a specific point within a game. We have obtained rudimentary play-by-play data from all 32 NFL teams over 7 seasons. The training data set consists of 183,361 data records with the following variables:

- Distance
- Down
- Play (Target Variable)

- Previous Play
- Quarter
- Score
- Season
- Time
- Yardline

The test set consists of 60,657 records. The target variable, "Play" has two states ("Pass" and "Run") that are present in a 1:1 ratio. This data set is thus reminiscent of data sets typical in financial trading applications where the target states are equally represented in the data.

### *4. Predicting Structure- (Oomycetes) Activity Relationships in chemical compounds:*

This data set represents a typical chem-informatics data set where there are many chemical descriptors that describe the structure of a chemical compound. The target feature is the presence of a desired bio-chemical activity (in this case against the fungicide Oomycetes) which is typically very rare. In this data set, there are 960 binary input features. The training data set has 107,440 data records with the target state activity: inactivity ratio being 1:175. The test data set has 35,862 records. This problem is thus representative of a high-dimensional needle in a haystack type problem.

## B. Summary of Analysis Methods:

For the comparative studies, we used the following methods from the Weka 3.6.2 Open Source Machine Learning Software toolbox:

1. Random Forest
2. Random Subspace
3. Random Committee
4. Random Tree
5. CART
6. ID3
7. Ensemble Selection
8. Bagging
9. Logistic
10. Bayes
11. Naïve Bayes
12. Classification by Clustering
13. Decision Stump

This provides a broad range of modeling methodologies against which we can compare our LeapWorks Predictive Analytics component. As part of the study, we also compared eight different flavors of LeapWorks Predictive Analytics:

1. Original genetic algorithm for tuple selection with 10 models – Using "Best Tuples"
2. Original genetic algorithm for tuple selection with 20 models – Using "Best Tuples"
3. New "no culling" (MTM) genetic algorithm for tuple selection with 10 models – Using "Best Tuples"
4. MTM genetic algorithm for tuple selection with 20 models – Using "Best Tuples"
5. Original genetic algorithm for tuple selection with 10 models – Using "All Tuples"
6. Original genetic algorithm for tuple selection with 20 models – Using "All Tuples"
7. MTM genetic algorithm for tuple selection with 10 models – Using "All Tuples"
8. MTM genetic algorithm for tuple selection with 20 models – Using "All Tuples"

Note: The key difference between the MTM genetic algorithm and our previous genetic algorithm relates to the selection of new parents within each generation. In addition, the term "Best Tuples" refers to building models with a subset of tuples that result in best performance on a tuning data set using a greedy selection approach. The term "All Tuples" refers to using all the tuples to define an individual model.

## C. Definition of Metrics for Comparison:

As discussed earlier, default values were used to set all parameter values for the Weka methods. In reviewing the Java docs for several of the best performing Weka methods, there does not appear to be any mechanism for tuning performance to maximize specific performance metrics such as minimizing the number of false negatives/false positives etc. The implicit assumption appears to be the minimization of total error.

For this reason, LeapWorks Predictive Analytics was run using default conditions (described in more detail in Section 4 and Appendix B) where the objective was the minimization of total error in the tuning data set. Furthermore, in view of the uncertainty around specific criteria used by the different Weka methods in building their respective models, the simplest and lowest order metric that we used for comparison is the area under the ROC curve:

From Wikipedia:

> "In signal detection theory, a receiver operating characteristic (ROC), or simply ROC curve, is a graphical plot of the sensitivity, or true positive rate, vs. false positive rate (1 − specificity or 1 - true negative rate), for a binary classifier system as its discrimination threshold is varied. The ROC can also be represented equivalently by plotting the fraction of true positives out of the positives (TPR = true positive rate) vs. the fraction of false positives out of the negatives (FPR = false positive rate). Also known as a Relative Operating Characteristic curve, because it is a comparison of two operating characteristics (TPR & FPR) as the criterion changes."

The advantage of using the area under the ROC curve as a first metric for comparison is that it does not assume a specific optimization criterion. If a specific optimization criterion is chosen, it amounts to choosing a specific operating point on the ROC curve. The total area under the curve can thus be viewed as a coarse overall performance metric of the modeling method, regardless of the specific optimization criterion that is selected.

One disadvantage of using the ROC Area alone as the comparison metric is that typically the end user has a specific optimization criterion in mind and thus operates at one point on the ROC curve. In view of these issues, we have also used the F-measure metric that represents the harmonic mean of precision and recall (see below) as a comparison measure between modeling methods. This is useful since there is typically an inverse relationship between precision and recall – eg. as precision goes up, recall typically goes down and vice versa. The F-measure is a lumped metric that includes both precision and recall measures to assess modeling accuracy.

From Wikipedia:

> "In statistics, the $F_1$ score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision *p* and the recall *r* of the test to compute the score: *p* is the number of correct results divided by the number of all returned results and *r* is the number of correct results divided by the number of results that should have been returned. The $F_1$ score can be interpreted as a weighted average of the precision and recall, where an $F_1$ score reaches its best value at 1 and worst score at 0.
>
> The traditional F-measure or balanced F-score ($F_1$ score) is the harmonic mean of precision and recall:
>
> $$F = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

Finally, in view of the high likelihood that the Weka methods are using minimizing total error as the optimization criterion, we will report Total Accuracy as one of the comparison metrics.

In summary, the three comparison metrics used in this study are:

1. ROC Area
2. F-measure
3. Total Accuracy

## D. Weka/LeapWorks PA Model Building Protocol

Significant effort was spent in developing automated protocols for performing the experiments on both the Weka methods as well as LeapWorks PA. The resulting scripts were all executed on the same computer described in Appendix A to remove hardware biases. In addition to quality of result metrics, model building times were

measured for all methods in order to generate the two dimensional quality of result versus performance curves.

Appendix A summarizes the testing protocol for running the Weka methods. Appendix B summarizes the specific default conditions under which LeapWorks PA was run.

## E. Summary of Results:

Comparative Studies of Modeling Methods against Unfiltered Data:
In order to benchmark the LeapWorks Predictive Analytics component against the set of Weka modeling methods, a detailed analysis was performed using unfiltered training data and unfiltered test data across the four data sets included in this study. This would allow a comparison of the modeling methods without including the effects of DUR filtering.

The accompanying Excel files (WekaUnFilteredSummary and PAUnFilteredSummary) provide detailed results across each of the four data sets. Here, we present some of the key results. In the charts below, the red diamonds refer to the different flavors of LeapWorks PA detailed in Section 2. In addition to the eight flavors outlined in Section 2, we added a ninth flavor of analysis (the MTM genetic algorithm for tuple selection with 20 models – Using "Best Tuples") where we skipped the Oomycetes run since the Weka Bagging method did not successfully complete the Oomycetes run. In some of the denser plots, not all the LeapWorks PA methods could be individually resolved in the plots. For this reason, most of the plots below are "zoomed in" views focusing on the optimal top left region of the plots.

1. ***Comparison of average out of sample ROC area (across the four data sets) vs Build Time:***

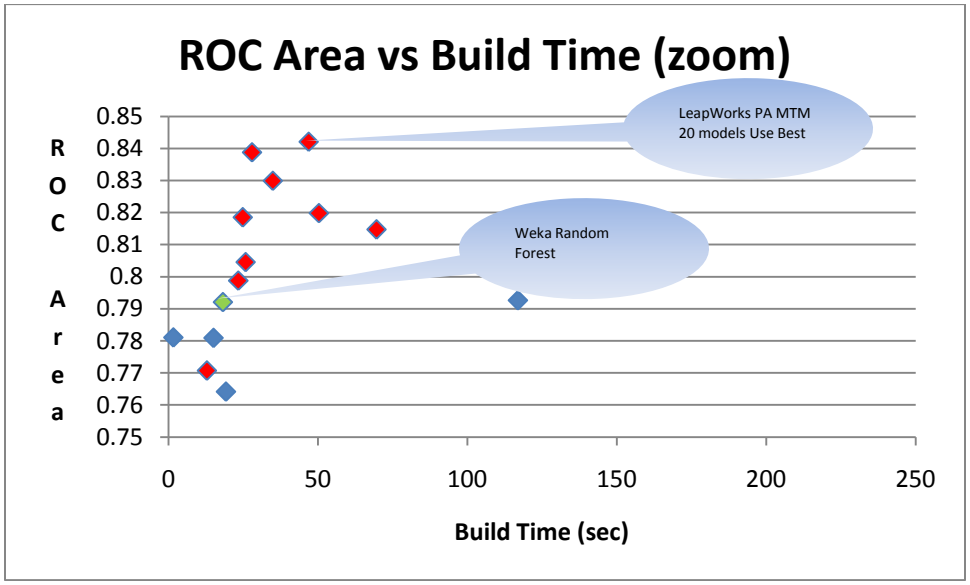


*Figure 16. <ROC Area> vs Model Build Time across all methods (zoom view)*

Figure 16 shows a zoomed in view of the <ROC> vs Build Time across all methods. The horizontal axis (Build Time) extends only up to 250 seconds to help resolve the fastest methods (as seen in Figure 17, the average build time for some of the Weka methods can be several thousand seconds). The clustering of the LeapWorks methods represented by the red diamonds at the top left corner of the plot indicates both excellent performance and robustness of LeapWorks PA across several different running conditions. The best Weka method in terms of both performance and <ROC> is the Weka Random Forest method indicated by the green diamond. It is a bit faster than LeapWorks PA with a lower <ROC Area> quality metric. We note that there have been significant performance enhancements in LeapWorks PA since these studies, and we will provide updated performance comparisons on an ongoing basis.

Figure 17 provides a "zoomed out" view of the <ROC> vs Build Time profile. As noted above, this figure provides a broader context through which LeapWorks PA performance can be assessed. The red diamond at the top left of Figure 17 indicates LeapWorks PA using Michael's GA running with 20 models and the Best Tuples per model. The yellow diamond on the top center-right indicates the Weka Random Subspace method that has comparable <ROC> but is ~50 times slower.

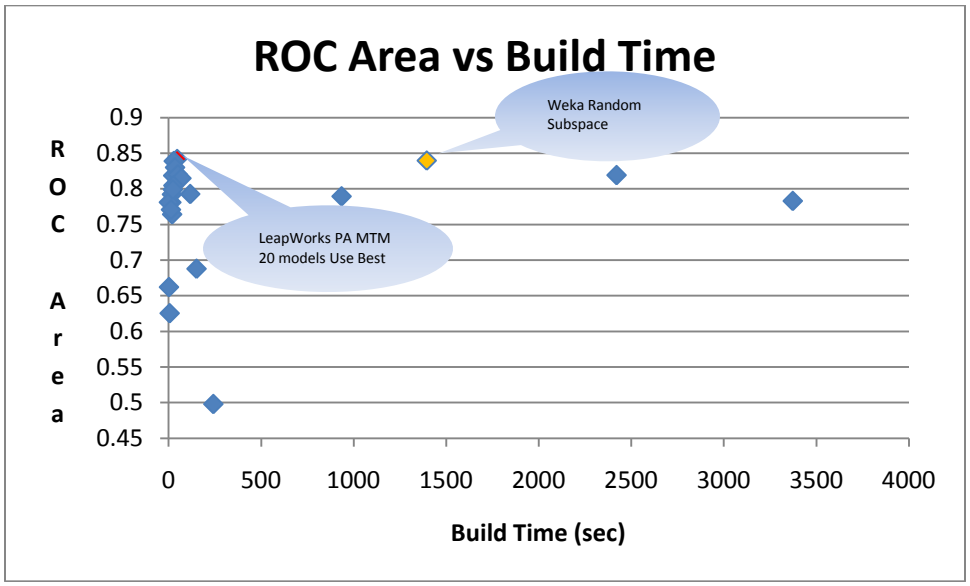


*Figure 17. <ROC Area> vs Model Build Time across all methods (total view)*

***2. Comparison of average F-Measure (across the four data sets) vs Build Time:***

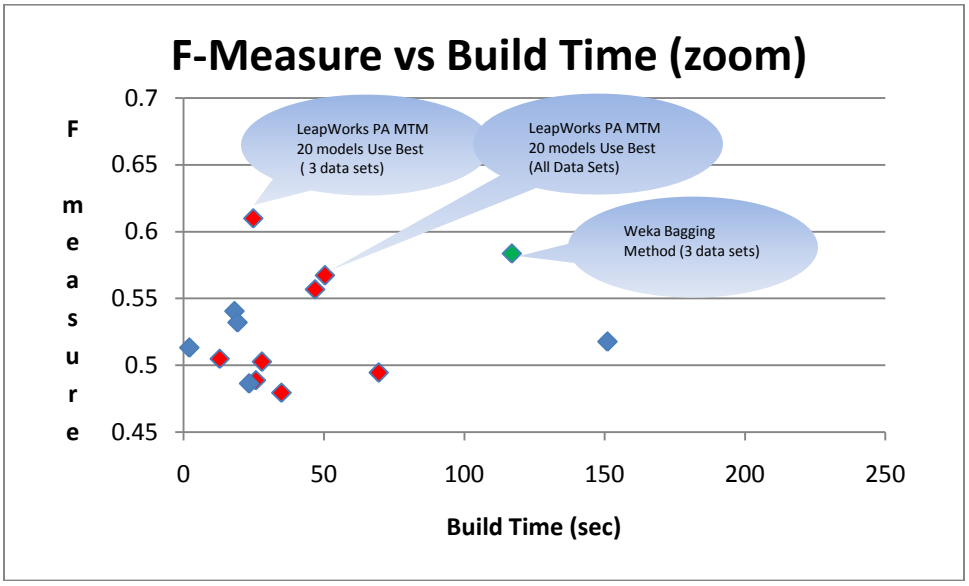


*Figure 18. <F-measure> vs Model Build Time across all methods (zoom view)*

Figure 18 shows a zoomed in view of the <F-measure> versus Build time across the data sets. It is interesting to note that the Bagging technique (highlighted by the magenta diamond) failed for the Oomycetes structure-activity data set with an out of memory error. Otherwise, Bagging was the best Weka method; for this reason, we have displayed LeapWorks PA across both the three data sets modeled by the Bagging method as well as across all four data sets.

***3. Comparison of average Accuracy (across the four data sets) vs Build Time***

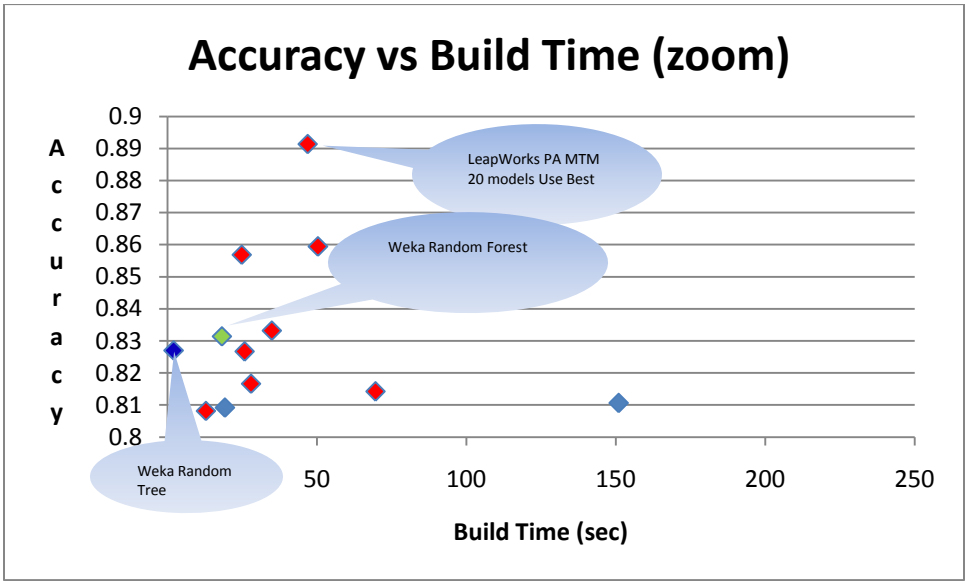


*Figure 19. <Accuracy> vs Model Build Time across all methods (zoom view)*

We note that LeapWorks PA has the best overall accuracy of all the methods. Weka Random Tree is very fast but with significantly lower accuracy. This is not surprising in that Random Tree builds only one tree model whereas LeapWorks PA is an ensemble based approach. As discussed above, we have made significant improvements to the performance of LeapWorks PA and updated reports will be forthcoming.

A point worth noting is that when we compare LeapWorks PA against Weka across the <ROC Area>, <F-measure> and <Accuracy> metrics discussed in this summary, different Weka techniques emerge as the best methods across the metrics. There does not appear to be one Weka method that is universally strongest across all metrics; this is not surprising in view of the diversity of the underlying paradigms. However, LeapWorks PA is consistently strong across all the metrics under a variety of running conditions. This is a testament to the robustness and general applicability of the technology.

### *4. Comparative studies of modeling methods on DUR filtered data:*

In an accompanying document (*DURImpactStudies)*, we have summarized more detailed studies on the impact of the LeapWorks Data Utility and Relevance (DUR) component on subsequent analysis. In this section, we present results on the following use cases described earlier in the Motivation:

- Unfiltered training data/DUR filtered test data
- DUR filtered training data/Unfiltered test data
- DUR filtered training data/DUR filtered test data

For simplicity of discussion, we present results on Weka Random Forest and LeapWorks PA MTM 20 models Use Best. Random Forest appears to be closest to LeapWorks PA in performance and quality of results. This is not too surprising given that both methods are ensemble based methods. In the tables below, the suffices A, B, C, D refer to:

| Suffix | Test Configuration |
|---|---|
| A | Filtered training/Filtered test |
| B | Filtered training/Unfiltered test |
| C | Unfiltered training/Filtered test |
| D | Unfiltered training/Unfiltered test |

### *5. Weka Random Forest:*

| Test Configuration | ROC Area | Accuracy | Precision | Recall | F-Measure | Lift |
|---|---|---|---|---|---|---|
| NFL_A | 0.706474182 | 0.661449 | 0.696566 | 0.706173 | 0.701336303 | 1.237471 |
| NFL_B | 0.642937699 | 0.595908 | 0.640587 | 0.598484 | 0.618820273 | 1.168815 |
| NFL_C | 0.705499511 | 0.659524 | 0.695365 | 0.703201 | 0.699261324 | 1.235339 |
| NFL_D | 0.658274274 | 0.619698 | 0.650818 | 0.66045 | 0.655598686 | 1.187483 |
| | | | | | | |
| Oom_A | 0.963107069 | 0.992739 | 0.765957 | 0.782609 | 0.774193548 | 48.15541 |
| Oom_B | 0.75089131 | 0.995957 | 0.757895 | 0.371134 | 0.498269896 | 140.1011 |
| Oom_C | 0.979909828 | 0.991874 | 0.777778 | 0.684783 | 0.728323699 | 48.89855 |
| Oom_D | 0.872269853 | 0.995985 | 0.727273 | 0.412371 | 0.526315789 | 134.4405 |
| | | | | | | |
| Intrusion_A | 0.786440764 | 0.618547 | 0.876737 | 0.620907 | 0.726971406 | 1.071952 |
| Intrusion_B | 0.744912106 | 0.625948 | 0.844698 | 0.653276 | 0.73675652 | 1.054218 |
| Intrusion_C | 0.870498053 | 0.702806 | 0.971175 | 0.656106 | 0.783138971 | 1.187417 |
| Intrusion_D | 0.859986647 | 0.715181 | 0.943694 | 0.685431 | 0.794091536 | 1.177769 |
| | | | | | | |
| Cancer_A | 0.799324947 | 0.967342 | 1 | 0.122066 | 0.217573222 | 26.88263 |
| Cancer_B | 0.686929781 | 0.994607 | 0.842105 | 0.054795 | 0.102893891 | 149.1882 |
| Cancer_C | 0.765160816 | 0.964373 | 1 | 0.042254 | 0.081081081 | 26.88263 |
| Cancer_D | 0.777945534 | 0.994935 | 1 | 0.10274 | 0.186335404 | 177.161 |

*Table 1: Weka Random Forest performance under different testing conditions*

We note that configuration A (DUR filtered training/DUR filtered test) results in improvement in the <ROC> and <F-measure> metrics over configuration D (unfiltered training/unfiltered test) in three of the four data sets. The intrusion data set is the exception and it may be related to our earlier observation that we have a true multi-state classification problem that we have mapped into a two state problem. However, the Accuracy metric is lower for Configuration A versus Configuration D in three of the four data sets. This variability needs to be studied in more detail and it may be due to the underlying signal to noise in the original data set. DUR filtering is most generally useful when there is significant noise in the data. In cleaner data sets, there may be more variability depending on the specific metric being assessed. The accompanying report (DUR Impact Studies) further details how filtering can be applied iteratively to build a hierarchy of models that result in improved performance.

### *6. LeapWorks PA MTM GA 20 Models Use Best:*

| Test Configuration | ROC Area | Accuracy | Precision | Recall | F-Measure | Lift |
|---|---|---|---|---|---|---|
| NFL_A | 0.708051742 | 0.697204 | 0.75555 | 0.683075 | 0.717487 | 1.342259 |
| NFL_B | 0.627673018 | 0.606047 | 0.707291 | 0.479726 | 0.571695 | 1.290523 |
| NFL_C | 0.69873861 | 0.693543 | 0.768788 | 0.651511 | 0.705308 | 1.365776 |
| NFL_D | 0.660615301 | 0.637981 | 0.69475 | 0.605493 | 0.647058 | 1.267641 |
| | | | | | | |
| Oom_A | 0.982779034 | 0.990145 | 0.701149 | 0.663043 | 0.681564 | 44.08096 |
| Oom_B | 0.765392526 | 0.995511 | 0.666667 | 0.340206 | 0.450512 | 123.2371 |
| Oom_C | 0.987026987 | 0.988762 | 0.631068 | 0.706522 | 0.666667 | 39.67497 |
| Oom_D | 0.91301662 | 0.995176 | 0.612903 | 0.293814 | 0.397213 | 113.2986 |
| | | | | | | |
| Intrusion_A | 0.899522399 | 0.837542 | 0.907553 | 0.892258 | 0.899841 | 1.10963 |
| Intrusion_B | 0.71442219 | 0.7612 | 0.884181 | 0.807778 | 0.844255 | 1.103495 |
| Intrusion_C | 0.954577455 | 0.889331 | 0.978145 | 0.884451 | 0.928941 | 1.19594 |
| Intrusion_D | 0.906582048 | 0.942825 | 0.947287 | 0.983363 | 0.964988 | 1.182254 |
| | | | | | | |
| Cancer_A | 0.862373102 | 0.965246 | 0.612903 | 0.178404 | 0.276364 | 16.47645 |
| Cancer_B | 0.848884848 | 0.994239 | 0.468085 | 0.150685 | 0.227979 | 82.92641 |
| Cancer_C | 0.784740123 | 0.879497 | 0.151825 | 0.488263 | 0.231626 | 4.08145 |
| Cancer_D | 0.888242744 | 0.989735 | 0.191214 | 0.253425 | 0.217968 | 33.87574 |

*Table 2: LeapWorks PA performance under different testing conditions*

The relative comparison of Configuration A versus Configuration D for LeapWorks PA follows a similar trend as for Weka Random Forest with one difference being the <ROC Area> metric for the Cancer Data. As noted earlier, this variability needs to be studied in more detail and again it may be related to the inherent signal to noise in the original data set. It is also important to note that LeapWorks PA has generally outperformed Random Forest under *all* the configurations. This provides additional support around the high quality and robustness of the LeapWorks Data Analytics platform.

## F. Discussion:

The results reported on in this summary across several diverse data sets have shown that the LeapWorks Predictive Analytic component compares very favorably with best of breed analytics both in terms of quality of results as well as performance. Significant performance improvements continue to be made that will further differentiate the LeapWorks platform.

The impact of DUR filtering on different data sets needs to be studied in more detail in the context of the inherent signal to noise in the original data set. As discussed above, DUR filtering is most generally useful in noisy data sets – this is where the key value proposition lies. Financial data sets are excellent examples of such a noisy data environment. In this study, the F-measure metric which includes both precision and recall elements showed the most consistent improvement when DUR filtering is used. Appendix C documents some recent Internet discussions on the superiority of Weka methods such as Random Subspaces and Random Forest. Such ensemble based methods represent state of art modeling methodologies. The favorable performance of LeapWorks PA as compared to these methods, as well as the enhancements described in our sister document on DUR Impact Studies, have positioned QLI's LeapWorks Data Analytics platform extremely well in the new landscape of Advanced Analytics that include cloud based analytics, in-database analytics and streaming analytics. Adapting our platform to a solution capability across different vertical markets can provide QLI with a distinct competitive edge as we move forward.

## 4.2 Multi-Agent Framework, Intelligent Agents, and Flexible User Interfaces

**Summary:**

A fundamental component in any organization is the support of human decision makers in visualizing the information and knowledge gained by the awareness component, in analyzing and deciding on plans of action, and in directing and monitoring operations in real time. In large organizations, these decision makers are distributed across time and space. The technologies developed to address these problems resulted in advancing capabilities for *Multi-Agent Frameworks, Intelligent Agents* and *Flexible User Interfaces*. This has resulted in improvements to the Hermes software components and libraries and development of the PARA (Pro-activity Amongst Rational Agents) software components and libraries.

The Hermes and PARA frameworks enable computing environments (agents) capable of operating pro-actively, reactively, and autonomously with regard to the tasks they are given and in collaboration with other agents, making decisions about their actions and prioritizing the actions to meet the requirements of their goals. In addition, these tools assist users in creating agents declaratively and at a higher conceptual level than is possible with traditional development, based on goals, rather than programmatically. Service discovery and integration is automated and performed ad hoc, without a-priori knowledge of the details of the service, semantic descriptions, and ontologies needed. This enables agents to search for, locate and invoke appropriate (orchestrate) the necessary sets of services needed to fulfill their current goals as well as a framework in which such semantic services can be deployed and discovered.

Additionally, the Hermes and PARA frameworks support the collaborative decision making process required to maintain control over the awareness and action aspects by users. They provide the capability to present dynamic information according to the goals that the user is trying to achieve. They support constructing both information and functional user interface visualizations dynamically as the inferred needs of the user changes when the execution of their goal unfolds (typically, goals within virtual enterprises are derived from the roles of the user).

Hermes and PARA technology continued to be developed as part of a joint effort between QLI and SAIC (Science Applications International Corporation) as part of the UICDS (http://www.uicds.us/) program. Even though UICDS is not part of this contract, we are including a description of the program as well as a summary of the advances developed as part of this program to demonstrate the capabilities of our Multi-Agent Framework, Intelligent Agents, and Flexible User Interface technologies.

The following sections describe Hermes, PARA, and a summary of the requirements, design, and integration with the Unified Incident Command Decision Support system in conjunction with SAIC.

**Hermes:**

Hermes is a framework for developing and deploying a service-oriented architecture built within an agent paradigm. It abstracts away the details of discovering, provisioning, and invoking services and provides a flexible internal agent architecture which is composed from reusable and interchangeable components. Hermes also provides a custom process (workflow) language that allows a developer to quickly compose new services, thus offering dynamic, ad hoc and opportunistic service reuse. Common agent-oriented interaction protocols, like subscription and contract net, allow for collaboration amongst agents and facilitate reasoning beyond those provided by traditional SOAs.

Some of Hermes' features include:

- Services are easy to implement and deploy, with simple APIs
- Automatic registration/deregistration of service descriptions with directory services
- Common interaction protocols for accessing services on remote agents and inter-agent collaboration
- Process support provides easy service composition within and across agents
- Support for common agent level protocols, like contract-net and subscription
- The internal functionality of an agent is comprised from its component set
- Components are reusable and interchangeable
- Message transport details are largely hidden from application-level code

**PARA:**

Agents are typically used in virtual enterprises, such as DoD command structures, to represent the goals to be achieved by the particular role each agent is assuming. PARA allows agents to reason about and prioritize the tasks necessary to achieve those goals which are often conflicting and need to be achieved in collaboration with other agents. As a result, these types of agents can be characterized as being rational (that is, their actions can be predicted and understood) and pro-active (that is, they take the initiative in achieving their goals). PARA is a multi-agent system in which rational decision making agents operate in an autonomous fashion to continuously discover and publish new information. These agents will be able to proactively report their results to entities which will trigger the decision making phase.

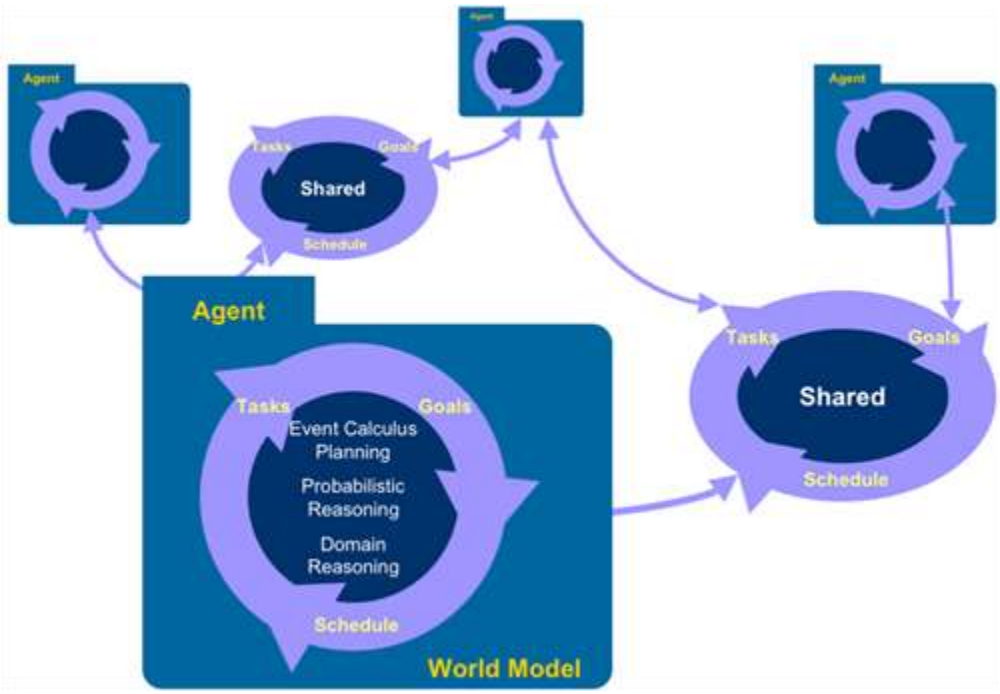


*Figure 20. Agent-based worldview and information shared with other agents*

**Project Accomplishments**

- Event-Calculus based reentrant planner
  - Embedded as a Hermes agent component
  - Based upon a Java implementation of a custom Prolog-like logic programming language
  - Goals can be planned for, committed to, executed, and cancelled
  - Goals have priority
  - Plans are partially-ordered abstract timepoints attached to actions
  - Goals can be bound to concrete times for planning
  - Actions have costs which can be dynamically calculated based upon a particular instantiation
  - Basic GUI for visualization of the planner and current agent state

- Agents have a set of goals, a current world state, and available actions
- Agents advertise goals which they can accomplish
  - Uses existing Hermes service directory with semantic action extensions
- Agents can contract out goals to be fulfilled by others
  - Contract winner determined by total cost of plan (currently time-based)

PARA is a high-level planning, reasoning, and decision making capability that runs on Hermes to adequately model complex, dynamic distributed systems. It is an Event Calculus based planning engine, which can be deployed as a component with a Hermes agent. The logic framework extends the standard SOA by wrapping the services of an agent with meaningful descriptions (metadata) that specify the service pre-conditions and effects within a formal model of the application domain. The driving force behind an agent's activity is a set of goals, which can be triggered by events in the agent's environment (sensor data, communications), or can be given by other agents or users. Based on the level of processing power and autonomy within the application domain, each agent is able to determine a sequence of actions to achieve their goals. Each goal or action can be related to information and resource requirements. A PARA-agent is able to constantly evaluate its environment and adjust its actions as a result of new events in line with its existing set of goals. PARA allows for goals to be prioritized and resources to be allocated accordingly. High-level cooperation is enabled by sharing goals and establishing commitments to goals among agents, which abstract out the details on how each agent achieves their goals and support is provided to execute shared plans. At the same time, PARA agents retrieve information and secure resources that are required to accomplish their goals.

**Standards Compliance**

- PARA is implemented using Java 1.5 or higher
- PARA is a Hermes component and relies on its APIs, in particular with regard to the task scheduler and the services definitions

**Event Calculus Planner**

- The heart of the PARA component is an event calculus based planner implemented in PARA's own version of Prolog on top of Java
- Event Calculus is a very well understood logical framework to express actions, the situations they can be executed in, and the effects of actions
- Within the Event Calculus planner, several different search strategies could be employed to find a plan
- PARA uses a bound iterative deepening strategy as its search

**Action Descriptions**

- Event Calculus uses a simple mechanism of associating logical conditions with timepoints representing the state before and after the execution of an action
- The state before an actions is executed is described by a set of *holdsAt(fluent, timepoint)* formulas
- The effects of an action are expressed as a set of *initiates(action, fluent)* and *terminates(action, fluent)* formulas

- Fluents are simple propositional logic formulas

**Composite Actions**

- From the definition of simple atomic actions, composite actions can be formed
- Composite actions can reference states across multiple timepoints inside the composite action, and, define constraints within those

**Central Schedule**

- The planner is focused around a single schedule for an agent, holding all executing and planned actions, and the state of the planner
- Agents can plan further actions within the already established schedule
- This facilitates the re-use of already scheduled actions within different plans, and the consistency of the overall agent behaviors

**Plan Cost**

- Each activity that PARA can plan over has an associated cost with it (possibly depending on the world state)
- In the simplest of cases, the cost represents the estimated time of execution of the action
- This allows the agent to compare different plans and pick the best one according to the cost function

**Re-Entrant Planner**

- As new information comes in, the planner constantly updates its state and the agent's schedule
- New information can be a new goal to plan on, or state information, which could trigger a re-planning for older goals that where using wrong assumptions
- A PARA agent constantly adapts its behavior to its environment

**Concrete Timepoints**

- PARA can reference real timepoints for goals, constraints and actions
- A PARA agent can define its schedule not only over a partially-ordered list of timepoints, but also reference concrete time
- Agents can agree upon a certain state at a certain time, thereby being able to plan for that condition without having to exchange synchronization messages (coordination with reduced communication)

**Goal Driven Plan State**

- Within the planner, all state information can be related back to the original source which is either a goal from within the agent, or a goal given by another agent
- The association of plan state with the original goals allows the planner to redo whole sections of the state if the underlying set of goals change

**Multi-Agent Planning**

- Agents can cooperate on a higher level by exchanging goals with other agents

- Exchanging goals relies only on a commonly understood model of the world and its states, as opposed to the intricate knowledge of the inner workings of the other agents that are required by standard SOAs
- An agent that gives another agent a goal does not have to know in what possible ways the other agent achieves the goal
- Contracting Goals
  - As each plan has cost associated with it, an agent can "contract out" a goal to a set of potential candidates
  - Each candidate gives the contracting agent an estimate of its plan cost for the goal and then the initiator can then select the best plan for its goal
- Commitments
  - An agent can be engaged with multiple agents at the same time
  - When an agent responds to a goal contract, it enters a commitment with the contracting agent to execute the plan
  - A plan is temporarily put on the agent's schedule, and thereby gets included in any other ongoing planning activity
  - If the agent loses the contract because the plan was too "costly", it not only eliminates the plan related to the contracted goal from the schedule, but also re-plans all the other goals that depend on some parts of the plan
- Integration with Special Reasoners
  - PARA employs an Event Calculus planner, which is in essence planning from first principles
  - While planners of this type can deal with a wide variety of domains, it is important to understand that many domains have specialized reasoning mechanisms, which operate far more efficiently in those domains
  - As a simple example, route planning, while possible within PARA's Event Calculus planner, can be done more efficiently using standard graph alorithms
  - PARA provides simple API's to include such special domain reasoners into the standard planning algorithm
  - As a general principle, the Event Calculus planner provides the agent with some baseline rationality, within which more specialized AI algorithms can be embedded

## UICDS Participation (Unified Incident Command Decision Support):

Hermes and PARA technology continued to be developed as part of a joint effort between QLI and SAIC (Science Applications International Corporation) as part of the UICDS (http://www.uicds.us/) program. Even though UICDS is not part of this contract, we are including a description of the program as well as a summary of the advances developed as part of this program to demonstrate the advances made regarding our Multi-Agent Framework, Intelligent Agents, and Flexible User Interface technologies.

**UICDS in Brief** (from http://www.uicds.us/files/UICDS%20in%20Brief.pdf )

Unified Incident Command and Decision Support (UICDS) is a national middleware framework to enable information sharing and decision support among commercial, academic, volunteer, and government incident management technologies used across the country to prevent, protect, respond, and recover from natural, technological, and terrorist events. UICDS is designed around data standards and the National Information Exchange Model (NIEM) to support the National Response Framework (NRF) and the National Incident Management System (NIMS), including the Incident Command System (ICS).

UICDS links homeland security and emergency management organizations, from incident command at the scene of an emergency to local and state operations centers to federal departments and agencies, from intelligence fusion centers to transportation management centers to health service organizations, and many other groups. UICDS is the standards-based middleware that exposes selected data from commercial and government applications and allows relevant emergency applications to subscribe to that information in order to have more than situational awareness – UICDS enables true sharing of information among applications so that each application's user can process, manipulate, expand, visualize, and share better and new information.

As middleware, UICDS does not interface directly with end users. Rather, it relies on regular, daily use, applications as the source of and visualization for relevant data. UICDS is the transporter of uniform data in common formats. Emergency applications (sensors, incident logs, personnel management, dispatch systems, video surveillance and intelligence tools – anything related to homeland security) provide a portion of their data to UICDS, which then publishes it to subscribers' applications. The applications then see the consumed data inside their own user interface. Thus, to the user, there is no new application, no new learning, and no conscious sending of information.

The portion of data obtained from applications is based on NIEM to compose a view of the incident. Such incident data also contain a link back to the original source so if someone wants the full details, they can link to external source data if they have appropriate permissions. Selecting the right information for delivery to the right person is accomplished by the UICDS Profile Service. The UICDS Agreement Service provides cross-agency and cross-jurisdictions information-exchange using the information equivalent of mutual-aid agreements.

UICDS accomplishes all this with a decentralized network of, perhaps, thousands of UICDS Cores with capabilities matched to end-user needs. For example, a large city,

state, or multijurisdictional region's UICDS installation may be a network of UICDS Core servers fully integrated with computer-aided dispatch, traffic sensors, hospital admissions systems, public works equipment maintenance records, arrest and warrant management systems, weather sensors, and more. Scale down UICDS to a single computer, lower communication bandwidth, add fewer external applications, and UICDS serves any type or size of community – urban or rural, coastal or desert, ski resort or football stadium, multiagency and multijurisdictional.

To date, more than 250 companies, universities, and government technology providers are engaged in UICDS. The technology providers participate in biweekly conference calls to discuss implementing and improving the UICDS services that support information sharing. This open setting has helped harmonize the UICDS data exchanges with the standards, data formats, and interfaces of current technology products to assure that technology providers can implement their UICDS adapters efficiently and with minimal effort. More than 125 organizations have downloaded the UICDS Development Kit which contains sample code and tools to build UICDS adapters for applications.

In 2009 a major demonstration of the UICDS prototype was hosted by the Virginia Division of Emergency Management to validate UICDS interfaces. The demonstration enabled information sharing among 23 applications across three jurisdictions. In 2010 the Federal Emergency Management Agency hosted an extensive demonstration of information sharing among applications used by federal, state, and local agencies in the National Capital Region. Throughout 2010-11, more than 100 pilots are being conducted in jurisdictions in more than 20 states to expand the number and type of technology provider applications that are integrated with UICDS and to further assure the proper use of UICDS in the field.

UICDS is being developed under a contract with the Department of Homeland Security.

**UICDS Requirements for Hermes and PARA**

1. User agents shall be available for the following types of user:

    a. Emergency Operation Center (EOC) director
    b. EOC HazMat specialist
    c. Neighboring EOC watch officer
    d. Incident Commander (IC)
    e. Law Enforcement
    f. Operation Manager

2. User agents in the agent network shall be preconfigured to receive notifications based on "interests" that match metadata of a work product.

3. User agents shall be able to receive notifications about new and updated work products.

4. User agents shall be able to visualize the following types of work product:

    a. New/Modified UICDS incidents

b. Incident Command Structure (ICS)
c. Maps
d. Map Layers (plume data, roadblocks)
e. Alerts (for Multi Agency Coordination System (MACS) personnel regarding potential EOC activation)
f. ICS 201 forms
g. ICS 203 forms
h. Population at Risk analysis and report data
i. Critical Infrastructure at Risk analysis and report data
j. Tasks lists sent from MACS

**UICDS Architecture Description**

The overall architecture for the UICDS system uses the QLI Hermes Multi-agent system to manage the users participating in the management of an incident and provide intelligent incident management support.

**User Management**

Each user is represented by its own User Agent on the machine or device of the user. The user agents communicate to the UICDS system via the agent network.

Access to the core services is facilitated by a Services Gateway Agent. The gateway agent uses a core's directory to determine the list of services, incidents, and resources. All access to core services is routed through the services gateway.

The concept of a User Proxy Agent is introduced to represent an end user to the core independent of the current network availability of the User Agent. It can cache important notifications to the User Agent while the User Agent is unavailable. Similarly, the User Agent can provide the user with limited interactions with cached UICDS artifacts while being offline.

The Registry Agent provides agents with yellow and white pages services. Additionally, since it is the first point of contact for an agent joining the Hermes system, it also provides login services for the User Agents.

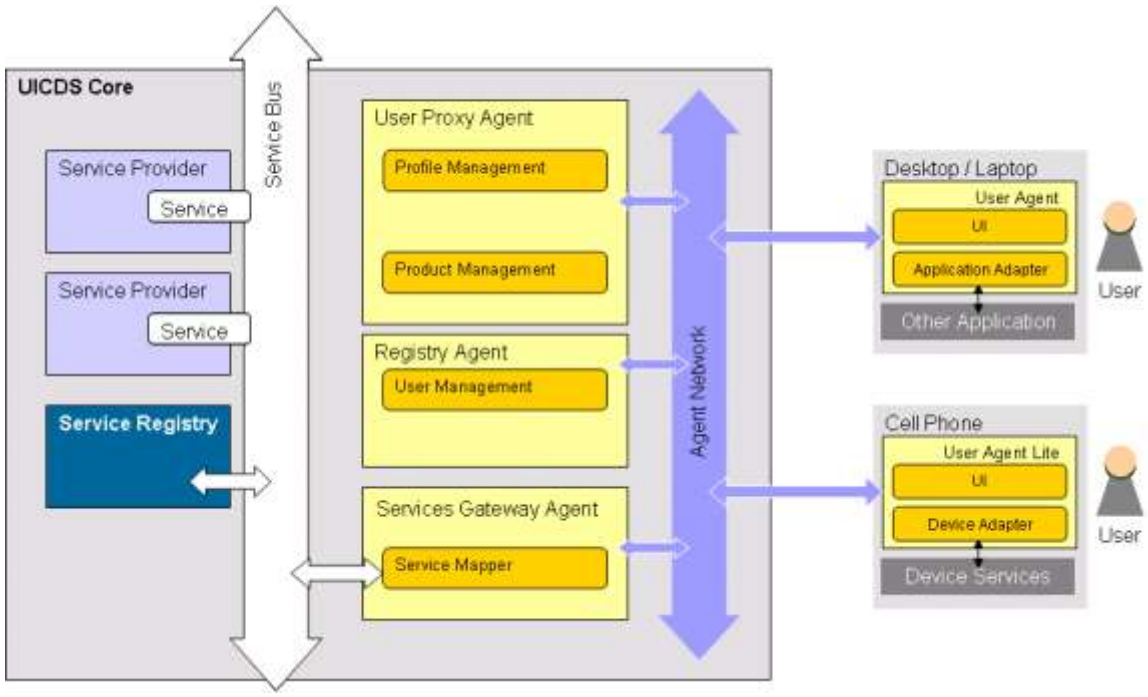


*Figure 21: Overall Architecture*

**Intelligent Incident Management Support**

As part of the set of core agents in the UICDS system, QLI provides a Core Coordination Agent (CCA), which provides "intelligent incident management" support services. The purpose of the CCA is to monitor the status of a core, the incidents within the core, and to initiate actions that affect the core or the user agents.

The CCA is represented with a system user profile in the core and receives notifications about status changes from the core through the standard core notification service via the Services Gateway Agent. The work product data from the core is translated into a logic representation in the CCA's knowledge base (KB). The knowledge base is implemented using QLI's proprietary implementation of PROLOG on top of Java.

The work product abstractions are defined as data records in the knowledge base. Furthermore, the KB provides some meta-predicates to define rules to handle changes in the core status and to define actions based on core services that can be triggered by the rules.

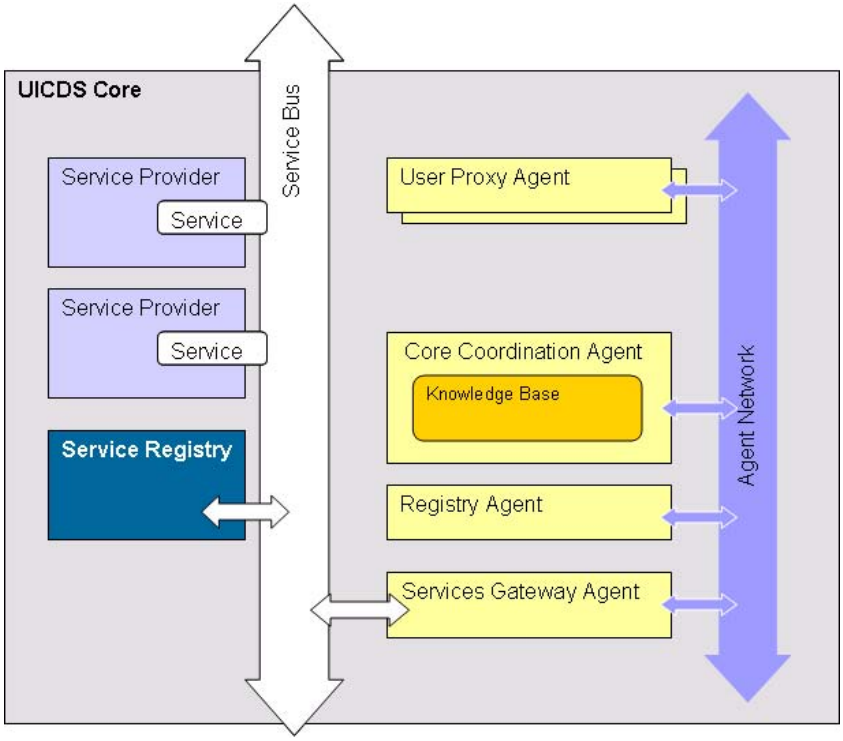


*Figure 22: Core Coordination Agent*

**System Overview**

The overall architecture is broken down in the systems and the events between the systems. The systems include the different agents (by type) and the capabilities of the core represented as a single entity. The system capabilities are provided by the Services Gateway Agent (SGA), a set of User Proxy Agents (UPA), and a set of User Agents (UA). The Registry Agent (RA) is required as part of the general agent system architecture to enable advertising and discovery services among the agents.

The SGA is built around a mapping component, which maps agent service requests to web service requests to the core. It can be viewed as the bridge between the agent network and the core.

The RA provides basic agent white pages and yellow pages services. Furthermore, the RA is used to link UAs to the system. UAs contact the RA with their user's login credentials, which the RA validates against the core and upon successful validation the RA creates a UPA as the user's representation in the core. The RA is also used to provide near real-time availability information about the UAs.

The UPAs provide the connection for the UAs to the core services and they are also designed to cache alerts to the user agent in case the UA is temporarily inaccessible over the network.

The system interactions can be defined by three categories of events:

- user driven events; this is the majority of the events in the UICDS system and they are handled by the user agents
- UICDS system events; these events include forwarding a request from the user to the core and notifying the user of an alert form the core
- agent system events; these are the events used within every agent system, including starting agents, registering agents, and finding other agents and their services

The core is designed to handle the various UICDS artifacts as work products. This simplifies the design of the system as the services gateway agent can handle most events by using the basic work product services in the core. The UICDS events between agents, the SGA, and the core are essentially making a service request related to a work product and receiving an alert from the core.

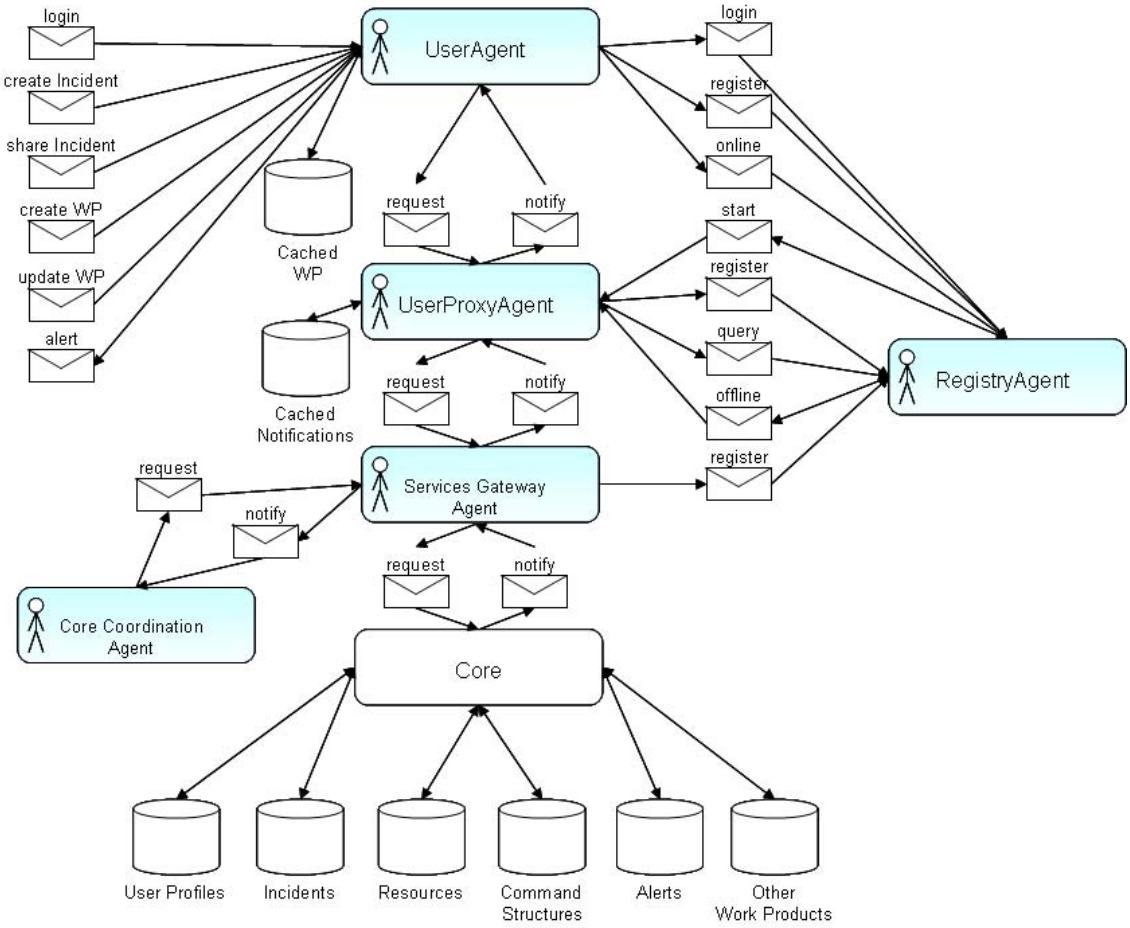


*Figure 23: System Overview Diagram*

### Deployment and Communication

The UAs are deployed on the users' machines or devices. The UPAs, SGA, and RA are ideally deployed in a separate process on the same machine as the core or at least within fast network reach. The core related agents all communicate directly in the same memory space.

Communication between the UAs and the core agents is facilitated by a secure HTTP agent protocol over a TCP transport protocol. This particular communication channel is designed to ease the installation behind firewalls. The RA is deployed with a component that facilitates connection management and message forwarding, called the Relay Server. The Relay Server accepts connections on a single port that can be opened and forwarded through a typical firewall setup.

The UAs and the RA connect to the Relay Server via its TCP port; they maintain a single, bi-directional connection. The Relay Server engages in a simple handshake protocol with a connecting agent and gives the agent its unique logical address.

When an agent sends a message to another agent, the Relay Server simply forwards the message.

### Blackberry User Agent

The UA on the Blackberry device requires special handling of communications, because RIM's Blackberry devices only support Java Micro Edition (CLDC Profile MIDP1.0/2.0), while the standard Hermes system requires Java Standard Edition 1.5 and higher. QLI has integrated the Blackberry devices by providing a special communications component. The Blackberry Bridge accepts incoming connections from the devices and handles the login of the device user. It provides forwarding of messages to and from the device to the Blackberry Integration component inside the respective UPA.

### System Subsystem Design Description

### HERMES Agents

Hermes agents are based on components providing the agent's functionalities. Each component can offer one or more services to the system, either within the agent, or to other agents. Each Hermes agent is built on top of a set of standard components.

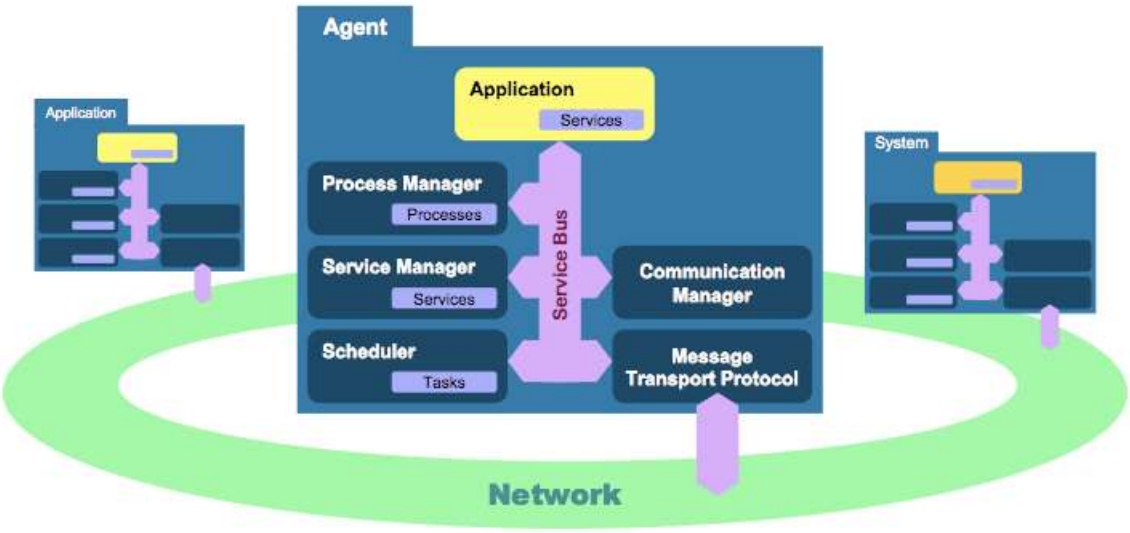


*Figure 24: Agent System – Agent – Component Decomposition*

**Generic Agent and Components**

Each agent is provided with a set of default functionalities that are implemented by a set of default components. The default functionalities include:

ServiceManager

ServiceManager is the interface for Service handling in this component. A ServiceManager handles the registering, deregistering and invocation of local services.

ProcessManager

The ProcessManager performs all process execution within the agent. All processes that are registered contain service endpoints that are provided to the ServiceManager. The Process-Manager provides several processes by default.

RegistryProxy

The RegistryProxy is responsible for providing registry services to an agent's components. These services include registering the appropriate services on the agent with the registry. The proxy looks for any service in the local ServiceManager that has been marked as a 'public' service, which indicates that it can be called from another agent. The proxy hides the details of registry communication from the rest of the components.

CommunicationManager

The CommunicationManager manages the details of routing all messages that come to/from the agent. This component manages the details of choosing an appropriate message transport for a message.

MessageTransport

A MessageTransport component is responsible for routing messages over an appropriate transport to another agent. Example transports include TCP sockets, HTTP, and internal to the VM communications.

**Services Gateway Agent**

Lifts UICDS Core Services to the agent level, provides notification endpoints to receive subscription data from the Core.

- cardinality: 1/UICDS Core; instantiated by Registry Agent on system startup
- functionalities: access UICDS Core Services, offer notification endpoint(s)
- data: WSDL service descriptions
- interactions with (which other agents/systems):
  Register Agent, User Proxy Agent, UICDS Core Services

ServicesGateway Component

The ServicesGatewayComponent is responsible for lifting the web services that the UICDS Core provides to the agent level. These services are registered within the Registry and available to the User Proxy Agent.

ImageServer Component

Component that renders map layers and returns an image.

Core Coordination Introspector Component

Component that handles allows the CCA Display Agent to introspect the CCA.

**CCA Display Agent**

- cardinality: as many as requested by users
- functionalities: visualizes CCA knowledge base internals
- data: Paralog facts, rules, actions
- interactions with (which other agents/systems): Register Agent, CCA

CoreCoordinationInterface Component

UI component for the Core Coordination Agent that shows clauses, rules and record types in the knowledge base as a graph. Rules blink when activated and records types show a detailed table of records of that type when clicked.

**User Proxy Agent**

Acts on behalf of the user within the UICDS system by tracking work products that are of interest to the user and forwarding those notifications on to the User Agent for display to the user

- cardinality: 1/user; instantiated by Registry Agent on successful user login.
- functionalities: subscribe to work products, retrieve work product, transform work product for device, transform work product for system, commit modified work product
- data: work products, user profile, incident work product, notifications
- interactions with (which other agents/systems):
  User Agent, Registry Agent, Services Gateway Agent

**UserProxy Component**

The UserProxyComponent manages all of the operations of the User Proxy Agent. It provides all of the services that are invoked by the User Agent to access the UICDS system. It also maintains the subscriptions to work products and handles work product notifications.

**ProxyOperations Component**

Performs the brunt of the UserProxyAgent's work. This component is used by the UserProxyComponent and the BlackberryProxyComponent to fetch and translate information from the core.

**Chat Component**

Component that provides the chat service. It provides a subscription to the user list and pushes messages on to the ChatInterfaceComponent in the user agent.

**BlackberryProxy Component**

The BlackberryProxy component is responsible for agent-to-blackberry proxy services within the UICDS agent system. Users on a Blackberry device can use their proxy to receive notifications.

**User Agent**

Provides a UICDS user access to the UICDS system through their device (laptop, smartphone, etc.)

- cardinality: 1/user device - created by user on device.
- functionalities: user login, get work product, view work product, modify work product
- data: device formatted work products, local device information
- interactions with (which other agents/systems):
  User, User Proxy Agent, Registry Agent, local device services

User Component

Component that provides the UserAgent with a connection back to the core agents. This component handles all of the details of logging in the user and listening for updates.

ChatInterface Component

Component that provides the client side of a chat service including UI.

User Interface Component

The User Interface Component provides the User Agent with a user interface for the user. The user interface includes the following capabilities:

- Online Status: show the user the current status
- Notifications List: present the user a quick overview of all current notifications and allow the user to retrieve the related work product
- Incidents List: allow the user to see the list of all incidents he/she is involved in
- Work Product Display: allow the user to view a work product in a manner suitable for the device; this is specific to the individual work product types
- Work Product Editor: allow the user to edit work products; the editor is device specific and also specific to the work product types

## 4.3 Medical Modeling and Situational Awareness

**Summary:**

QLI has developed a framework (Gryphon) for performing flexible, computationally efficient simulation and visualization of complex adaptive systems realized through the dynamic interaction of multiple modeling components. Gryphon has been used successfully in several real time exercises such as Cobra Gold 2008; Operation Caring Response to aid the humanitarian response to Cyclone Nargis (hurricane and epidemic disease outbreaks); and most recently as a primary tool to assist USNORTHCOM and HHS in modeling and managing the impact of the spread of novel 2009 influenza A (H1N1).

**Motivation:**

For decades, governments have been supporting development of technologies that harness computing capability to better support the health and prosperity of their citizens. Since the 2001 outbreak of foot-and-mouth disease in the United Kingdom and the 2003 global SARS outbreak, modeling and simulation platforms for infectious diseases have become increasingly valuable decision support tools in the US and UK. The successful applications of computing technology in 2009 H1N1 pandemic and other disease outbreaks have shown that timely response to disease outbreaks has become critical in a globally connected and resource constrained society where non-local disease spread can occur at increasingly faster rates.

**Individual-based models:**
Various highly complex individual-based models have been developed to understand and predict the spread of infectious diseases and the impact of treatment and control strategies. These individual-based modeling efforts have been supported by the National Institutes of Health (NIH) Models of Infectious Disease Agent Study (MIDAS) Program and various DoD programs since 2004. While individual-based models can capture the spread of disease with high-fidelity (including daily activities and connections of individuals via transmission networks), and can answer many scientific questions about the spread of an infectious disease, the complexity of the model makes it impractical for quick what-if analyses of interventions or treatments under different conditions.

Recent reports have demonstrated significant performance improvement for baseline simulations without interventions, where a general re-sampling algorithm was developed to model the spatial transmission of infectious diseases. For example, the team in Taiwan applied the algorithm to an individual-based stochastic disease simulation and showed how the location of the initial seed can influence the spatial spread of the epidemic in Taiwan. However, several fundamental issues remain to be addressed for individual-based models:

- Individual-based models cannot be easily supported by current surveillance data, whose resolutions are usually at zip code level or higher (e.g., county) in the US. It would be challenging to use these models as real-time decision support tools for policy assessment.

- Individual-based models are ideal for modeling local spread of disease within a county, but disease spread is often non-local, for example, across different counties in a state like Maryland, or between other states in the U.S. and Maryland in the DC metro area.
- New re-sampling algorithm greatly improves the run time of baseline individual-based simulations, but the run time of such simulations with interventions may still need hours or days for a country with millions of residents.

**Hybrid agent-based models:**

In addition to individual-based models, the US government has been supporting the development of structured population models for infectious disease and biological attacks, in both the NIH Programs (e.g., Prof. Gary Smith at the University of Pennsylvania, Prof. Alessandro Vespignani at Indiana University) and the Office of Naval Research (ONR) Integrated Warfighter Biodefense Program (IWBP). Built upon structured population models, QLI has been developing Gryphon, a hybrid agent-based multi-scale stochastic modeling and simulation platform for characterizing the geographic spread of infectious disease and modeling the effects of various mitigation strategies in a GIS environment.

Figure 25 summarizes the state-of-the-art modeling and simulation technologies for infectious diseases. Different modeling strategies are compared using both response time and complexity of the model. Gryphon integrates agent-based modeling with a stochastic structured-population susceptible-exposed-infectious-recovered (SP-SEIR) model. This hybrid approach provides several advantages over each pure method by combining rich modeling capabilities of agent-based modeling and low computational overhead of differential (or difference) equations. Therefore, Gryphon enables multiple rapid what-if analyses to be performed using singular or multiple interventions and allows users to optimize pandemic responses. Compared to recent efforts on equation-based infectious disease modeling for structured populations, Gryphon can support more complex user interactions and population behavior modeling of social groups for both disease modeling and interventions at runtime.

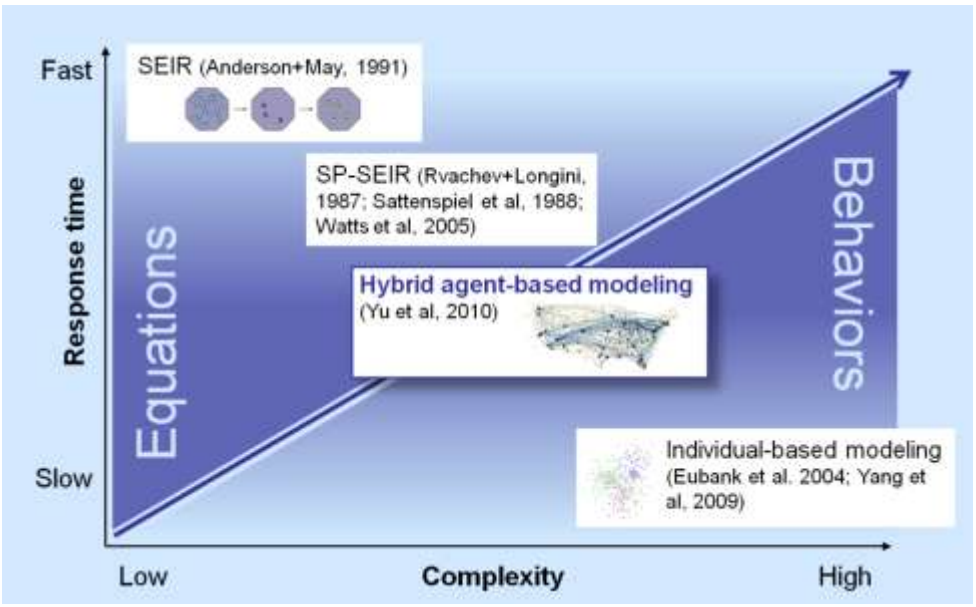


*Figure 25: Hybrid agent-based stochastic modeling, simulation and analysis platform*

Figure 26 describes the existing software architecture of Gryphon where a model manager maintains the models before/after a simulation step (e.g., saving data into database, responding to user input). The database shown is used to store the simulation data for replay, analysis and visualization.

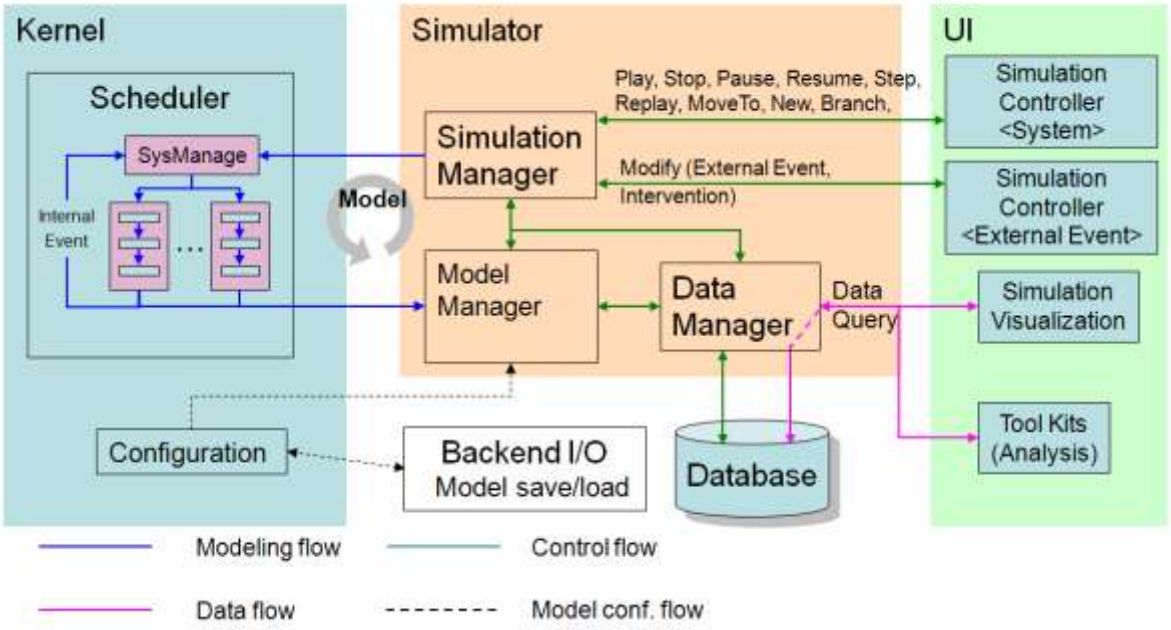


*Figure 26. Existing software architecture of Gryphon*

## Methods:

### Hybrid Agent-Based Modeling

A group of individuals associated with a geographic location (e.g., a country) is modeled as a primary group agent. A primary group agent can be decomposed into several secondary group agents. Each of the secondary group agents can be further decomposed into multiple tertiary group agents. Translocation is the process of decomposing each primary group into various secondary groups and populating locations with the corresponding secondary groups. The mixing of secondary groups at a location can be localized mixing or non-localizing mixing. Localized mixing refers to the manner in which members of all secondary groups at a location interact with one another. Non-localized mixing is the manner in which members of secondary groups at different locations indirectly interact with one another or with environments to spread disease such as indirect transmission of cholera via water. In this paper only localized mixing is considered.

Different from equation-based models such as SP-SEIR, the hybrid agent-based model does not have a migration matrix to determine the mixing rates among different groups. Instead, the mixing process is naturally driven by the behaviors of different groups. The behaviors of an agent include two parts: active and reactive. Active behaviors of an agent are modeled by a set of decision rules such as movement patterns, condition-based behaviors caused by interventions and environmental changes. The reactive behaviors of an agent in the context of infectious diseases refer to localized and non-localized mixing for a location, where the numbers of individuals at different disease states change constantly due to the interaction with other agents at the location.

Each simulation time step consists of three steps in the order of pre-step, step, and post-step. In pre-step, a secondary group agent may change its behaviors in response to either interventions or environmental changes. In step, secondary group agents at a location mix with each other based on a given disease model. In post-step, the system will update the state of each secondary group agent based on the calculation of the disease model. Subsequently, each secondary group agent notifies its primary group agent of the state changes. At the end of post step, all secondary groups at each location are cleared and the translocation process of each primary group agent is executed to prepare for next simulation time step.

### Disease Model

We use a discrete-time stochastic susceptible-exposed-infectious-recovered (SEIR) model to simulate the localized mixing of all secondary group agents at a location, where S(t), E(t), I(t) and R(t) represent the number of susceptible, exposed, infectious, and recovered individuals, respectively, at a location at time t. The total population at the location $N(t) = S(t) + E(t) + I(t) + R(t)$ is assumed to be a constant (birth and death are ignored). Specifically, the stochastic SEIR model is specified by the following difference equations.

$$S(t + h) = S(t) - B(t)$$

$$E(t+h) = E(t) + B(t) - C(t)$$
$$I(t+h) = I(t) + C(t) - D(t)$$
$$R(t+h) = R(t) + D(t)$$

where *h* represents the time interval between two continuous simulation steps and h is set to 1 day.

$B(t)$ is the estimated total number of infections resulting from individuals in the $I(t)$ state. For a given infectious person, the number of new infections from $N(t)$ is sampled from a binomial distribution as $M(t) = Binomial(Binomial(Poisson(c), S(t)/N(t)), p)$, where $c$ is the mean number of daily contacts per person and $p$ is the probability that a contact produces infection. Given $M(t)$, $B(t)$ can be calculated as the sum of $M(t)$ for all individuals at I state. Since a Poisson distribution is a special case of a Binomial distribution, the compound distribution $Binomial(Binomial\ (Poisson(c), S(t)/N(t)), p)$ can be reduced to $Poisson(\beta IS(t)/N(t))$, where $\beta$ is the transmission rate and $\beta = c \times p$ [9]. The number of individuals becoming infectious C(t) in a day can be represented by a binomial distribution $Binomial(E,\alpha)$, where $1/\alpha$ is the length of the mean latent period. Similarly, the number of recoveries $D(t)$ can be represented by $Binomial(I, \gamma)$, where $1/\gamma$ is the length of the mean infectious period. The values of mean latent period and mean infectious period are 0.85 day and 2.95 days, respectively. The daily transmission rate $\beta$ is estimated from the basic reproductive rate $R0$ as $\beta = R0 \times \gamma$.

**Validation studies:**

We studied the effectiveness of Gryphon, an agent-based stochastic simulation engine for infectious diseases using the historic SARS data. The estimated pairwise value of R0 for Hong Kong is consistent with Wallinga et al 2004 by assuming an exponential increase in the number of cases over time, while the predicated total case number for non-local disease transmission is close to the one given by Hufnagel et al 2004, in which Hufnagel et al. used a continuous-time stochastic SEIR model. The experimental results suggest that the expected numbers of infections as well as the timeline of enforced control strategies predicted by our stochastic engine are in reasonably good agreement with previous approaches.

In this validation study we simply use a pairwise R0 to capture the control strategies deployed upon the first and second WHO warnings.We can see that the peak of the simulated data from Gryphon drops very fast. This motivates us to develop the next generation of Gryphon technology for data-driven stochastic simulations, where the basic reproductive rate R0 is dynamically changing based on the available data during a disease outbreak. The data-driven Gryphon will serve as a real-time epidemiological environment for pandemic preparedness and response planning.

**Data Sets**

Important events in the timeline of the 2003 SARS epidemic in Hong Kong and other Asian countries are as follows

- February 15, 2003: Official report of a 33-year male and a 9 year old son in Hong Kong with Avian influenza (H5N1).
- March 12, 2003: First global alert about atypical pneumonia in Vietnam and Hong Kong was issued by World Health Organization (WHO).
- March 15, 2003: Second global alert about name of SARS and case definition was issued by WHO.

One simple way to model the transmission dynamics and control strategies is to change the basic reproductive rate R0. Therefore, instead of using one value for R0, we have a pairwise value (RH,RL) for R0 to reflect the level of effectiveness of control strategies after WHO warnings. The value of R0 is switched from RH to RL in the experiments based upon one of the WHO global alerts. The range of RH is {2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5} and the range of RL is {0.6, 0.7, 0.8}.

The travel data sets are generated from the International Air Transport Association (IATA) (http://www.iata.org) database, which contains the number of available sets between any two given countries. The country data sets, including population, latitude and longitude for each country, are generated from the website (http://www.geonames.org).

**Parameters for Local Transmission Dynamics**

The first experiment studies the basic reproductive rate for local transmission dynamics in Hong Kong. Figure 27 describes the cross-correlation coefficient between the mean of the simulated data for 100 rounds and the WHO data for different pairwise R0 values, where we seed two infected individuals in each simulation. According to theWHO data, there are only two infections in Hong Kong on February 15, 2003. Note that the two data series in Figure 27 are aligned to calculate the maximal cross-correlation coefficient between the simulated data and the WHO data. We can find that, for Figure 27, the cross-correlation coefficient is consistently higher when R0 is switched on March 15, 2003. This indicates that those serious control measures such as quarantine and isolation are implemented in Hong Kong only after WHO issued the second global warning on March 15, 2003.

However, it is hard to find the proper value of R0 only from Figure 27. The reason is that cross-correlation coefficient only models the shape of two data series. As we can see from Figure 27, it is difficult to tell whether pair (3.0, 0.7) is better than pair (3.5, 0.7) on modeling the spread of SARS in Hong Kong. One idea is to use accumulative case numbers as the second measurement to model the scale of the curves. Figure 28 describes the difference of simulated accumulative case# and actual accumulative case# for pairwise R0, where the value of R0 is switched on 3/12 and 3/15, respectively. From Figure 28, we find that the accumulative case number for (3.5, 0.7) is much closer to the WHO data based on the second global warning. The experimental results show that the combined two metrics, cross-correlation coefficient and cumulative case number, can effectively estimate R0 values from temporal patterns in an observed epidemic.

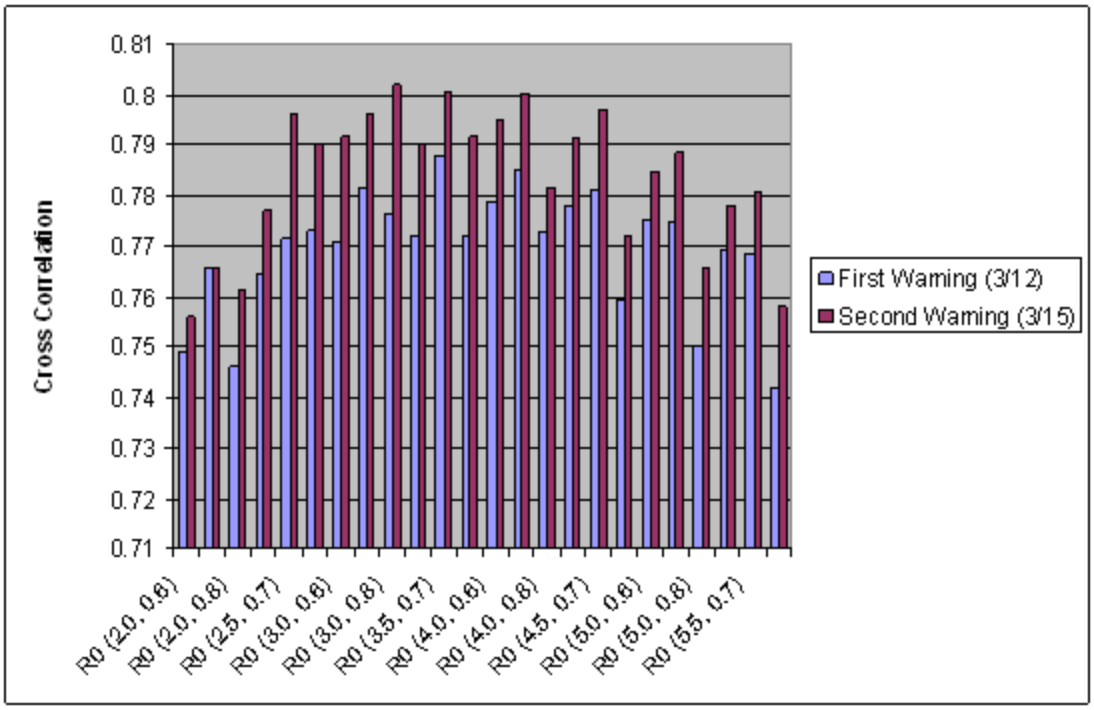


*Figure 27. The cross-correlation coefficient for different pairwise R0, where the value of R0 is switched on 3/12 and 3/15, respectively*

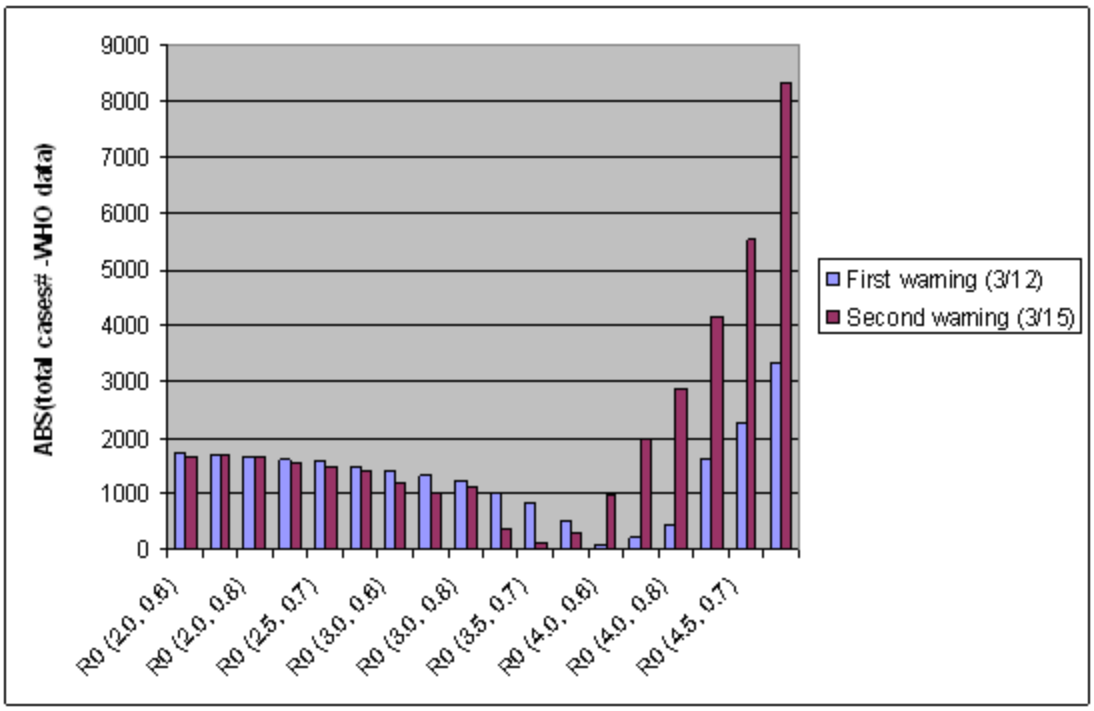


*Figure 28. The difference of simulated accumulative case# and actual accumulative case# for pairwise R0, where the value of R0 is switched on 3/12 and 3/15, respectively*

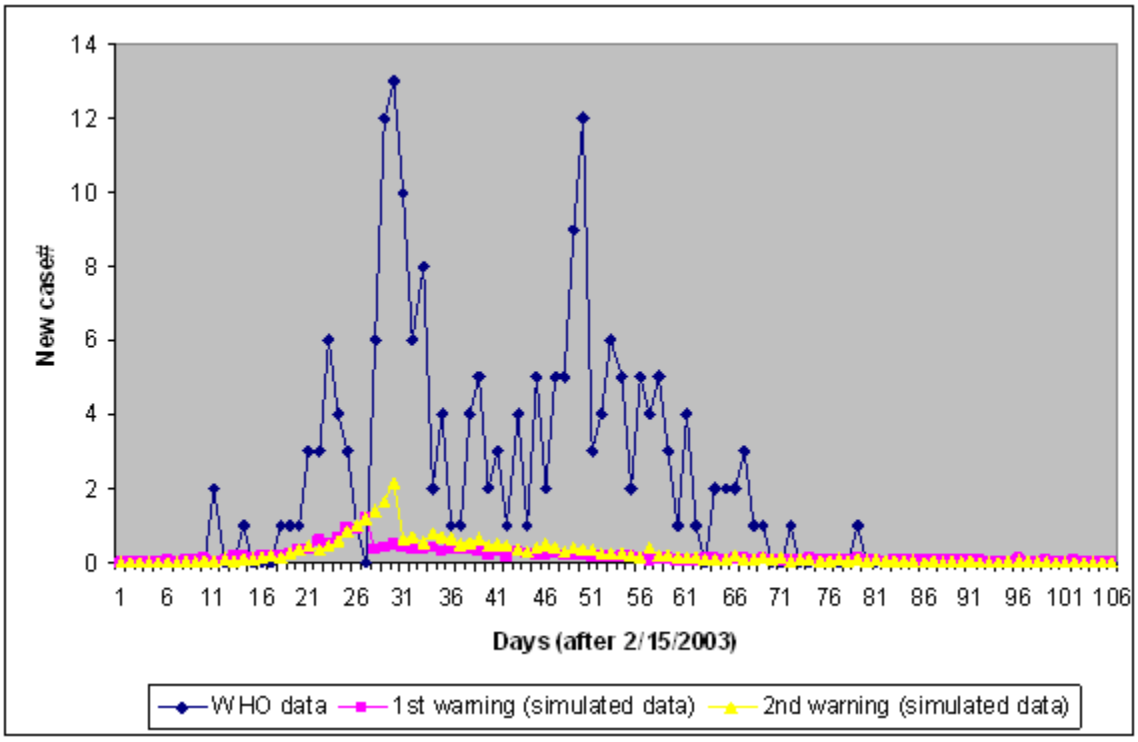


*Figure 29. The new case# of SARS in Singapore from WHO data and the mean new case# of SARS from the simulations (100 rounds) for the pairwise R0 at (3.5, 0.7)*

**Parameters for Non-local Transmission Dynamics**

The non-local transmission dynamics for SARS at the country level may depend on two factors: the airline travel and the probability that a sick individual travels from his home city to other cities. In this experiment we seed two infections in Hong Kong on February 15, 2003 and we examine the disease outbreak in two Asian countries: Singapore and Japan. We assume that the probability that a sick individual with SARS travels is 0.5. Figure 29 shows the new case# of SARS in Singapore from the WHO data and the mean new case# of SARS from simulations. We can see that, based on the airline travel data and the probability that a sick individual travels, the stochastic simulation engine significantly underestimates the SARS outbreak in Singapore. The peak of the mean new case number is one for the first warning and two for the second warning.

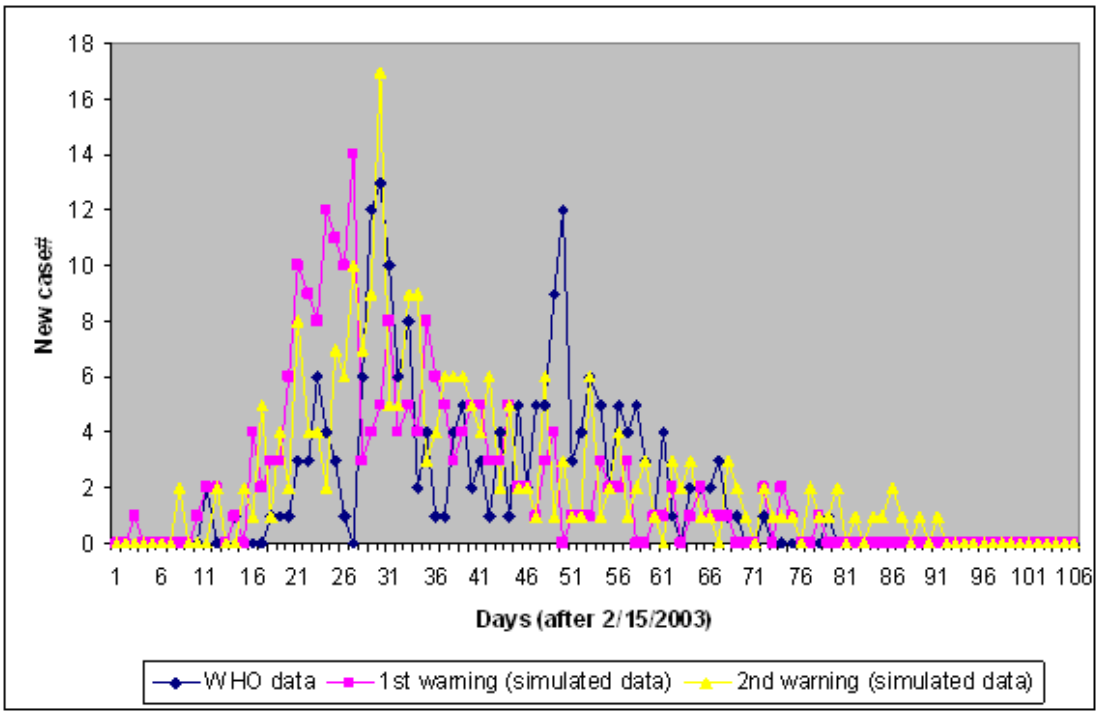


*Figure 30. The new case# of SARS in Singapore from WHO data and the best match new case# of SARS from the simulations for the pairwise R0 at (3.5, 0.7)*

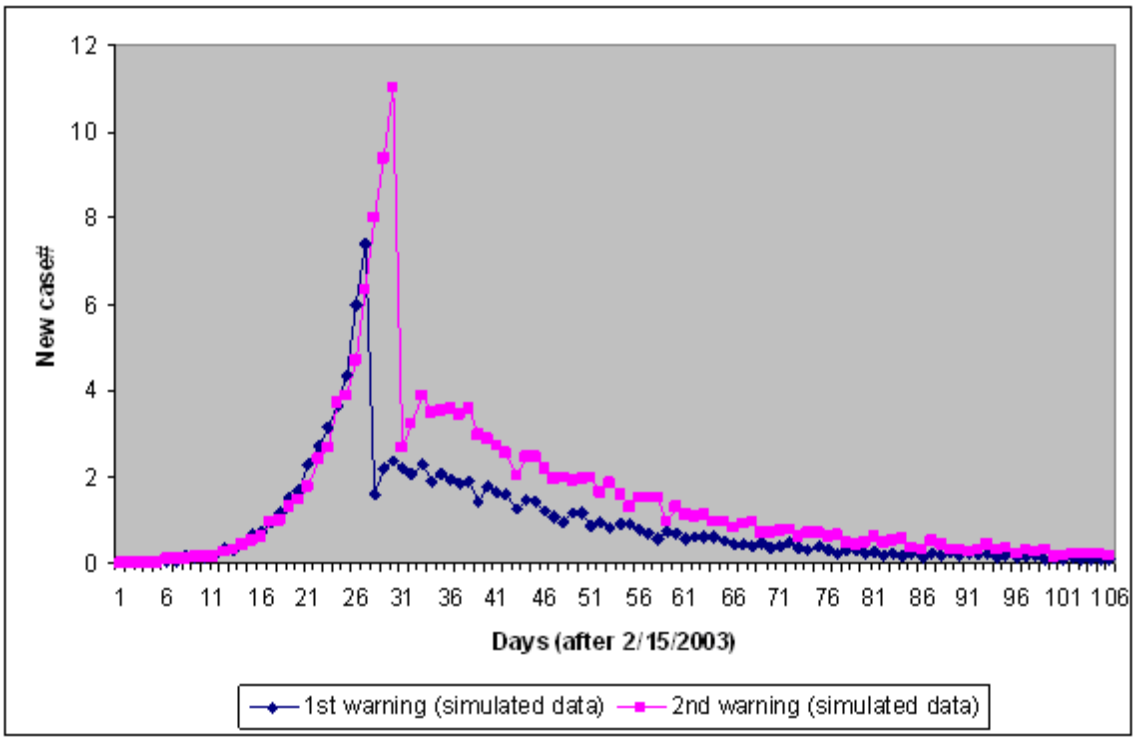


*Figure 31. The mean new case# of SARS in Japan from the simulations (100 rounds) with the pairwise R0 at (3.5, 0.7)*

However, some specific simulations did capture the dynamics of the SARS outbreak in Singapore. Figure 30 describes the best matched epidemic curve in Singapore from stochastic simulations. The cross correlation coefficient between the best matched simulation data and the WHO data is around 0.6. This indicates that the spread of the SARS epidemic in Singapore is probably the worst case scenario predicted by stochastic simulations. Most likely the first few infections are super-spreaders (a super-spreader is a person having many contacts) and they transmit the disease to large numbers of people.

As shown in Figures 29 and 30, the mean new case number predicted from the stochastic simulation engine can overestimate the SARS epidemic. On the other hand, the stochastic engine can also underestimate the SARS epidemic in some countries like Japan in 2003. The daily airline traffic (in 2005) between Hong Kong and Singapore is 38000 seats/per day, while the daily airline traffic between Hong Kong and Japan is 133000 seats/per day. Based on the traffic data as well as the potential of disease transmission from mainland China, the estimated SARS epidemic should be more severe in Japan than that in Singapore.

Figure 31 shows the mean new case# of SARS in Japan generated by 100 rounds of simulations. The stochastic simulation engine predicts that there will be about 98 SARS cases for the first warning and 160 SARS cases for the second warning in Japan. This value is close to the mean total case number of SARS predicted by the continuous time stochastic model published in Hufnagel et al 2004. However, there is no reported SARS case in Japan during the 2003 SARS outbreak. The actual scenarios in Singapore and Japan motivate us to rethink more complex processes of the spatial and temporal transmission as well as different modes of transmission. These include but not limited to the role of the super-spreader and the heterogeneous mixing among different social groups. For example, Japanese tourists may not well mix with the Chinese community in both Hong Kong and mainland China. The high standard of hygiene conditions in Japan may also prevent the spread of SARS.

**Success Stories:**

**Operation Caring Response in Myanmar**

On May 2, 2008 Cyclone Nargis, the second deadliest named cyclone of all time hit Myanmar. Millions of people were affected by the storm. It was estimated that one million people would die without aid from other countries. US Pacific Command embarked on a humanitarian mission to help those affected by Cyclone Nargis. As part of this mission the need arose to assess the threat of disease outbreak in Myanmar and dynamically simulate the impact of both Medical and nonmedical interventions. The Gryphon Infectious Disease Simulator was a natural fit for these challenges. Pre- and post-cyclone population data for refugees, civilians, military, and non-governmental organizations (NGOs) were gathered from WHO reports and other sources. Cholera, Pneumonia, Malaria, and Hepatitis A were selected as representative diseases as their methods of transmittal are typical of diseases encountered during relief operations. For each disease, several interventions suggested by domain experts have been incorporated. Examples include vaccines, antibacterials, hand washing campaigns, and insecticide-treated bed nets.

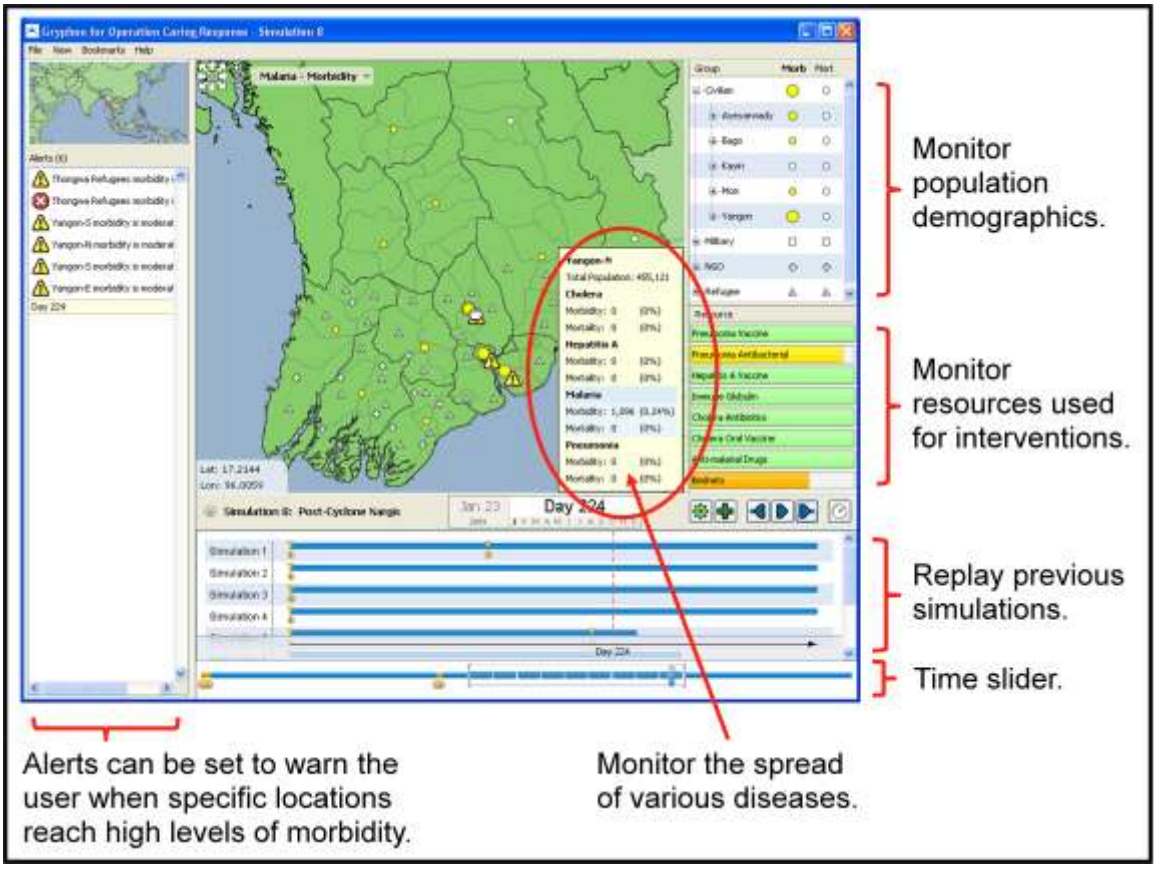


*Figure 32. Gryphon GIS features – Operation Caring Response example.*

Accomplishments: The following enhancements were added to Gryphon in 12 weeks:

- the ability to simulate multiple diseases simultaneously.
- models of Myanmar both before and after the cyclone.
- four new diseases and seventeen corresponding interventions.
- the ability to model the seasonality of diseases.
- the ability to export data to CSV.
- the ability to add and configure resources used by interventions.
- updated web service interface for third party software.
- many user interface improvements.

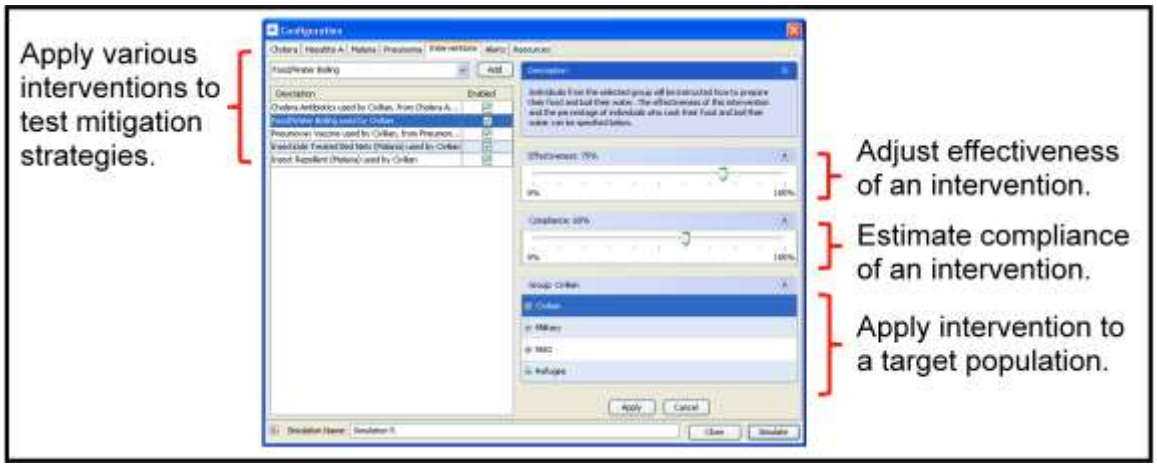


*Figure 33. Gryphon intervention configuration features.*

Results: In early September 2008, Gryphon Operation Caring Response was successfully delivered to Alion Science and Technology Corporation who will transition it to USPACOM. USPACOM will use this technology to help with subsequent natural disasters. Alion is currently pursuing transitioning Operation Caring Response into the U.S. military.

**Novel 2009 Influenza A(H1N1) and US NORTHCOM**

In March 2009, the US identified its first cases of 'swine flu' in New York. Initial reports of cases in Mexico, where the disease is thought to have emerged, indicated Novel 2009 influenza A(H1N1) was highly contagious, highly virulent, and showed high mortality. Decision makers were under great political pressure to design policies to prepare for and respond to this ongoing outbreak. In late May 2009, the Gryphon Infectious Disease Simulator was employed by the Commander, US Northern Command, to formulate nationwide response. Several medical and non-medical interventions were examined including antiviral treatment, masks, travel restrictions, and school closures.

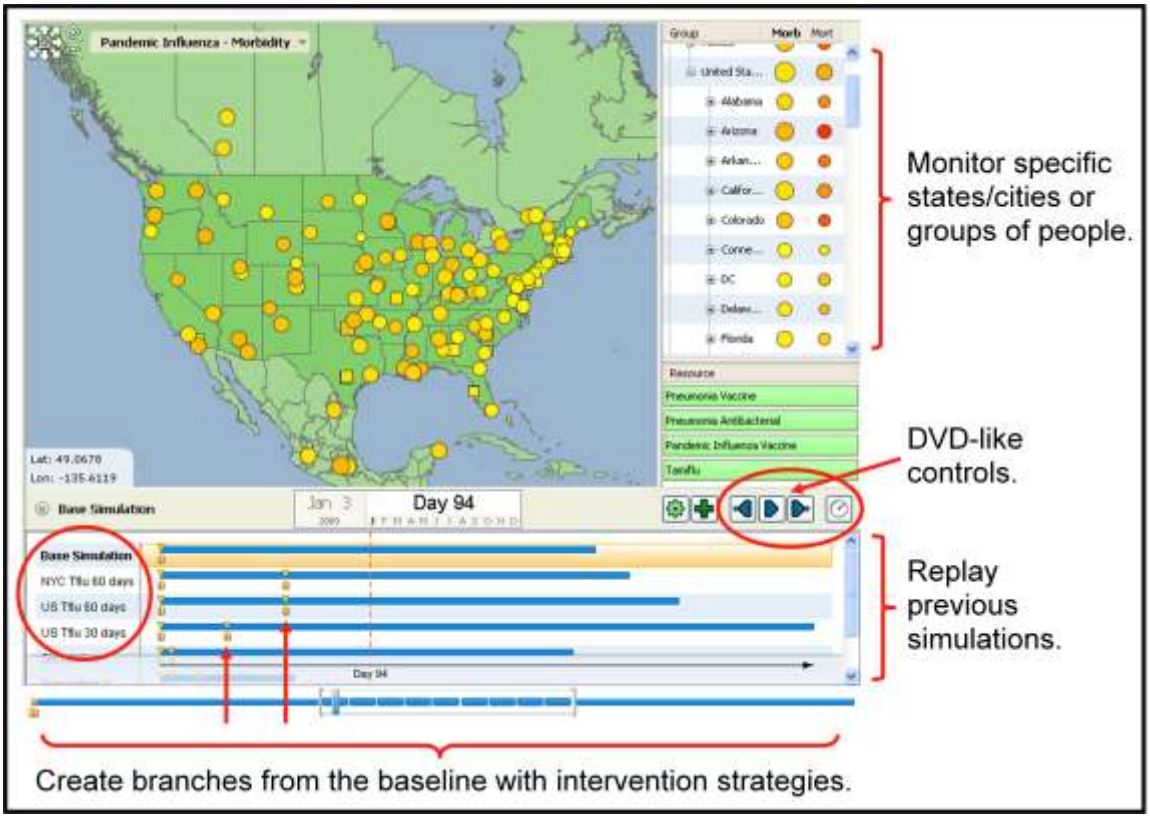


*Figure 34. Gryphon GIS features – USNORTHCOM example.*

Accomplishments: The following enhancements were added to Gryphon during April and May, 2009:

- the ability to process real-time surveillance data from CDC.
- the ability to study airline travel restriction and closing the border between the U.S. and Mexico.
- the ability to model vaccines, antiviral therapies and face masks at high fidelity.
- the ability to run stochastic simulations.
- the ability to deliver results within 24 hours.

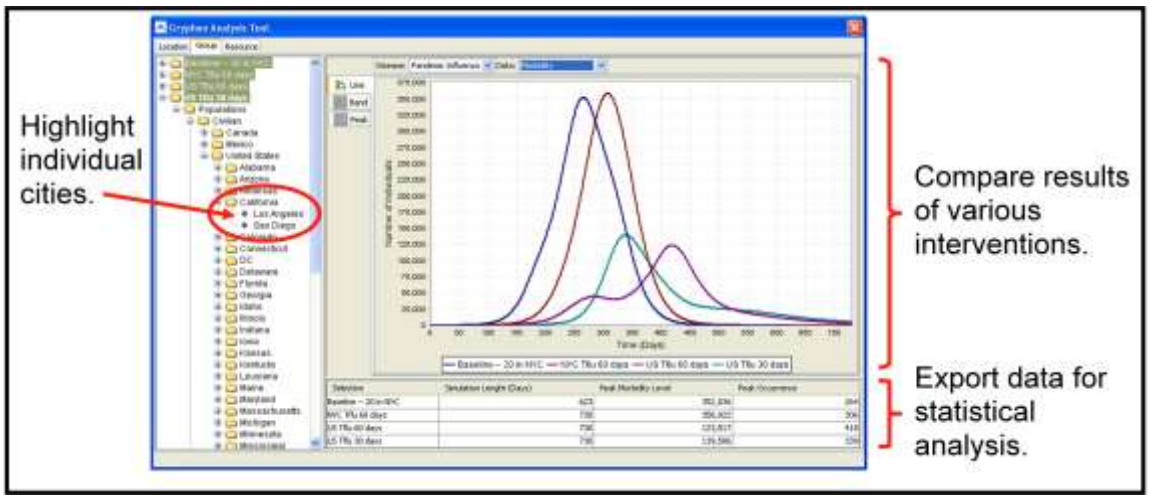


*Figure 35. Gryphon charting analysis features.*

Results: After CDC and HHS evaluation of available modeling and simulation software, Gryphon was the only non-government solution selected to support policy decisions in federal response to Novel 2009 influenza A(H1N1). Gryphon technologists were tasked nightly with various intervention scenarios, data were gathered overnight, and a report provided the following morning. Policy makers identified Gryphon's developers as "an S&T company that delivers."

## GryphonCloud

GryphonCloud is a secure Internet service allowing public health professionals to run simulations within their area of responsibility. GryphonCloud expands the scalability of Gryphon by modeling the state/county/installation level with high fidelity while maintaining surrounding states/regions with low fidelity. Users have great flexibility to create personalized intervention strategies based on local resources.

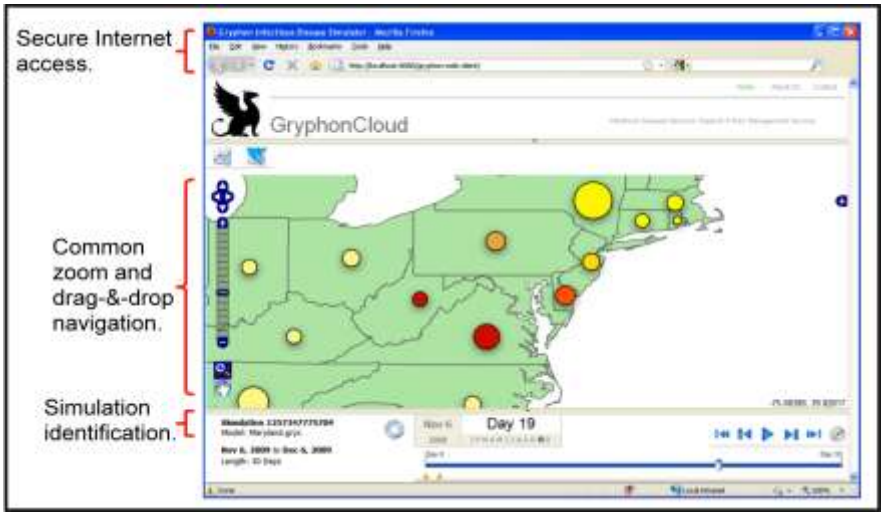


*Figure 36. GryphonCloud – GIS visualization.*

Currently, GryphonCloud focuses on tracking and simulating 2009 H1N1 influenza outbreak, but its underlying hybrid technology has been used to model other infectious diseases, such as pneumonia, norovirus, malaria and cholera. The GryphonCloud engine combines standard epidemiological models with Quantum Leap's agent-based simulation engine.

As a decision-support and resource management tool, GryphonCloud can reduce the risk and uncertainty surrounding emerging infectious diseases by allowing public health professionals to examine disease spread and test mitigation strategies within a simulated population. To start, the base simulation can be seeded and configured according to the user's knowledge of the outbreak situation and their geographical regions. GryphonCloud's unique user interface and architecture allows the user to examine a simulation in detail with DVR-like controls. In addition to running a base simulation, various intervention strategies (both medical and non-medical, alone or in combination) and assumptions can be simulated and tested. The base simulation can be compared and analyzed at any point with other simulations based on the tests and change in assumptions; the comparisons will provide insight as to what the effect of those interventions and assumptions will have on the outcome.

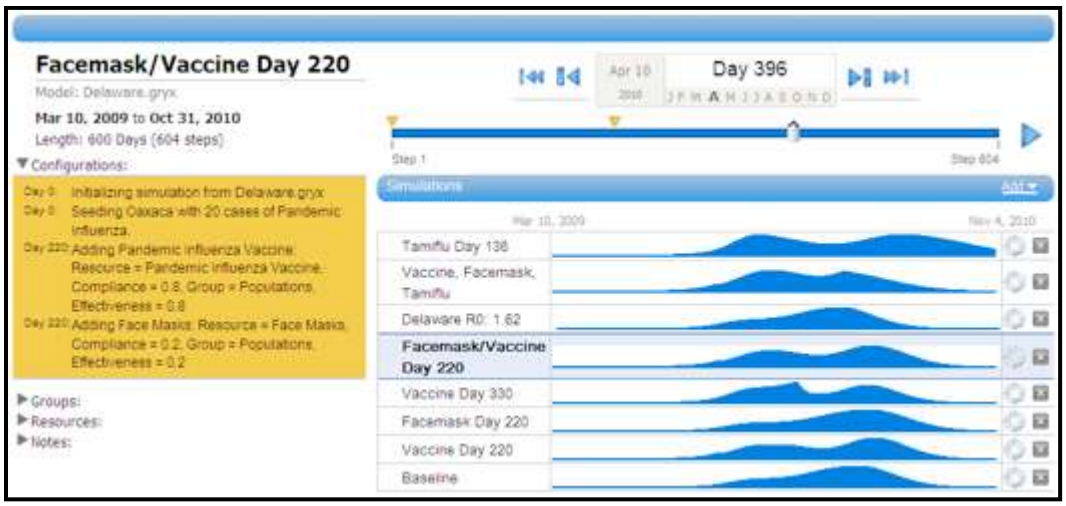


*Figure 37. GryphonCloud – simulation features visualization.*

The power of GryphonCloud lies in letting public health professionals quickly and reliably assess the benefit of specific interventions or treatments by dynamically invoking, measuring, and testing the impact of intervention strategies at any point during a simulated disease outbreak. Public health officials can use GryphonCloud to provide enhanced decision support by configuring models and interventions (including increasing fidelity for their areas of responsibility, combining interventions over the course of an outbreak, and changing disease attributes). Minutes later, the official will receive prognoses that will enable them to inform the best courses of action. Because GryphonCloud permits multiple tests to be set up and performed rapidly using singular or multiple interventions and treatments, each user can optimize his/her guidance and actions for effective disease management, as well as determine resources required for treatment, including personnel, and to provide advice about the potential impact of the

disease to support decisions of government authorities regarding forecasting, planning and response.

Interventions which GryphonCloud currently supports or will support in the near future include antivirals, vaccines, face masks, and social distancing policies. Successful mitigation strategies will result in the flattening and lengthening of the baseline epidemiological curve as shown in the graphical visualization below. Furthermore, access through GryphonCloud.com will give users opportunities to collaborate with colleagues across the nation and provide a connected, universal understanding of disease outbreaks and success levels of potential mitigation strategies.

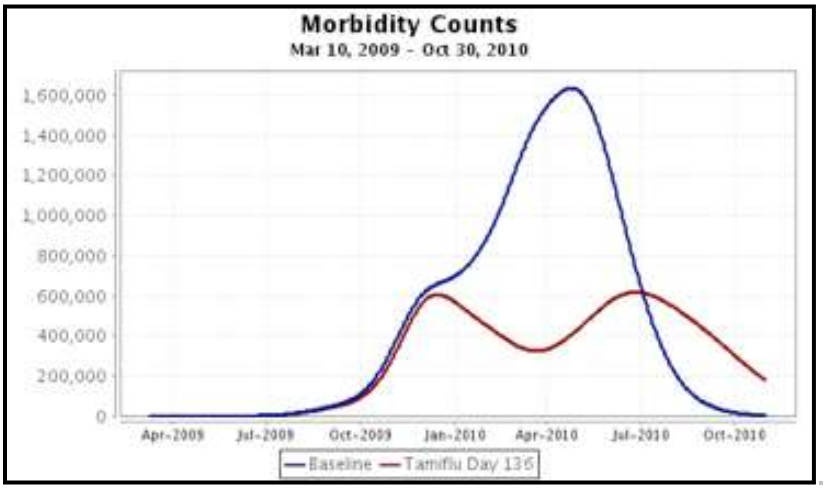


*Figure 38. GryphonCloud – charting analysis features.*

Highlights of GryphonCloud include:

- Providing DoD-proven Gryphon technology to public health officials in states, cities, regional "all-threat" facilities via secure connection on the Internet
- Providing public health professionals the ability to manage, view, and understand disease spread in their regions for multiple scenarios, focusing on "in your backyard" analysis
- Enabling in-silico hypothesis testing through “what-if” analysis
- Enabling public health officials to manage infectious disease with actual available applicable interventions for a specific geographic area
- Providing a scalable "cloud" computing architecture to handle spikes in usage caused by multiple users during disease outbreaks and seasonal influenza

Users interact with the GryphonCloud service via:

- An "epi-curve"-based visualization
- A map-based interface to playback and review simulated outbreaks
- Detailed reports comparing simulations of outbreaks
- Real-time, real-world data visualization and integration

In addition to the 2009 H1N1 influenza spreads, GryphonCloud will in the future be used for:

- Other emerging infectious diseases
- Chronic disease modeling in populations
- Modeling impact of theoretical interventions

# Appendix A: Weka Testing Protocol

***Performance testing environment***

- Windows 7 operating system, 64-bit
- Java version info – 1.6, 64-bit Java Virtual Machine
  - java version "1.6.0_20"
  - Java(TM) SE Runtime Environment (build 1.6.0_20-b02)
  - Java HotSpot(TM) 64-Bit Server VM (build 16.3-b01, mixed mode)
- Used Weka 3.6.2
- Command line JVM arguments: -Xmx4096m -XX:+UseConcMarkSweepGC
- Hardware
  - Processor: Intel(R) Pentium(R) D CPU 3.00GHz
  - RAM: 4GB
  - More detailed information is available at http://share/public/Administration/Systems/Inventory/Reports/Tahoe.htm

***Weka analytics performance testing methodology***

- Used a jython scripting environment (version 2.5.1) for testing automation; used as glue to connect test steps
- Steps for each classifier, data combination
  - Loaded training instances, measured load times
  - Built classifier based on loaded training instances, measured classifier build times
  - Created instances from testing data and evaluated the model/classifier built on the training data
  - Recorded evaluation and timing results
- Main script logic is below (only main flow logic included):

```
startTime = time.time()
trainingData = Instances(FileReader(File(training_data_location)))
trainingData.setClassIndex(trainingData.attribute(target_feature).index());
loadEndTime = time.time()
loadTime = loadEndTime - startTime

classifier.buildClassifier(trainingData)
classifierBuildTime = time.time() - loadEndTime

testingData = Instances(FileReader(File(test_data_location)))
targetAttributeIndex = testingData.attribute(target_feature).index()
testingData.setClassIndex(targetAttributeIndex)
targetPositiveIndex = \
        testingData.attribute(target_feature).indexOfValue(target_state)

eval = Evaluation(trainingData)
eval.evaluateModel(classifier, testingData, [])

testResults = TestResults(eval, targetPositiveIndex, loadTime,
        classifierBuildTime)
```

## Appendix B: Operating Conditions for LeapWorks PA – email from 8/19/2010

Folks,

I've been chewing on the best/fair way of running LeapWorks PA for our initial validation studies. The intent of this first set of studies is to get a broad overview of how we stack up against other methods. As Michael has pointed out, in any specific problem, the domain analyst will tweak to get the best results for the metric (eg. lift, recall etc) that they are interested in. We are running Weka with default settings on each method. As such, the fair thing to do is to set up reasonable defaults for our LeapWorks PA for these studies.

After thinking about this a bit, here are my recommendations – they are a bit different from the original settings in my summary a few months ago based on some more experience using our tool:

1. On the training data, use a 90% training and 10% tuning split for all datasets.
2. Use the Michael McGowan default dimensionality that is used as the default dimensionality in our current builds to run the tests ( I finally decided against looping through several dimensionalities to find the best dimensionality for the initial tests, as this is somewhat similar to optimizing settings in Weka that we did not perform).
3. Use all 1D features to drive GA ( in most cases this will be 100; for NFL it would be 8).

4a. Run with current GA with default settings for GA parameters ( 20 generations, cull rate 0.6, mutation rate 0.01).

4b. Run with Michael's new GA : NEED TO CALIBRATE MICHAEL'S NEW GA BASED ON HIS RECOMMENDATIONS

5. Select # tuples per model to be 10 (downgraded from 20).
6. Select # feature sets to be 1.
7. Select # models per feature set to be 10 (downgraded from 20). Note: We ran with both 10 and 20 models

8a. Use All tuples for Model Building

8b. Use Best tuples for Model Building with Minimize Error and 0% tolerance

9. For tuning models, use Minimize Error as the evaluation algorithm.
    - Note that in step 4, I would like to run LeapWorks PA using our current GA as well as Michael's new GA for comparison purposes.
    - Note that in step 8, I would like to run model building with All tuples and Best tuples since I am not sure yet about the differences between the two model building approaches. You should include processing times for both 8a and 8b, and report results for both 8a and 8b. I believe that generally, Use Best will improve Precision and reduce Recall, but we need to document this during our studies. I have used Minimize Error for the Best Tuples mode as I am not trying to tune a specific performance.

I think this is a clean way to run LeapWorks PA. Bert, you can run this script using 4a (our current GA) right away while Michael is working on his new GA.

Thanks,

Ganesh

## Appendix C: Internet Commentary on Weka Random Subspace Method

Random Subspace Method
by Colin Bick Dec 11, 2009; 12:26am

Hi all,
I've been having unexpected success using the Random Subspace classifier. I am hoping somebody can point me towards some good references, or describe personal experience, explaining what properties of a dataset might cause this combining technique to greatly outperform others.

Appreciated,
Colin

---

Re: Random Subspace Method
by Peter Reutemann-3 Dec 11, 2009; 12:34am

Tin Kam Ho (1998). The Random Subspace Method for Constructing Decision Forests. IEEE Transactions on Pattern Analysis and Machine Intelligence. 20(8):832-844. URL http://citeseer.ist.psu.edu/ho98random.html

Cheers, Peter

Peter Reutemann, Dept. of Computer Science, University of Waikato, NZ
http://www.cs.waikato.ac.nz/~fracpete/ Ph. +64 (7) 858-5174

---

Re: Random Subspace Method
by Harri Saarikoski-2 Dec 11, 2009; 07:27am

hi, my personal experience comes with bioinformatics datasets, protein recognition

we / multitude of others experimented with (not random but) manually selected subspaces where selection was carried out on the basis of maximal similarity between features
(-> where maximally similar features formed one subspace)

results were better than with the full space of features, as you here find with random subspaces and was earlier found with random forests and rotation forests

reason why it works is that such random/explicit subspaces provide 'views' into the data that happen to be more homogeneous than with the original full space of features i.e. subspacing, even random, removes noise from the data by removing noisy (as well as useful) features from the original set

iow, it works for the same reason as feature selection/reduction methods work and are subject to the same reservations concerning their overfit tendency: be advised e.g. that to a certain (probably predefinable) extent random selection leads to random performance against testsets beyond the training set and is subject to the random seed parameter used to shuffle features->subspaces and sensitive to the (fixed) number of features included in each subspace

this can be partially fixed by having a very high number of trees/forests to avoid the bias (narrowing of 'view') resulting from subspace selection

Harri

## Appendix D: Glossary

**Computationally efficient**: Use of a computer system, having one or more processors or virtual machines, each processor comprising at least one core, the system comprising one or more memory units, one or more input devices and one or more output devices, optionally a network, and optionally shared memory supporting communication among the processors to produce the desired effects without waste.

**Data Management**: The organization of data typically provided by a database management system.

**Data Storage**: The storage of data typically within a database.

**Data support discontinuity threshold:** A discontinuity threshold in the filter union data support used as a pre-filter to select a filter.

**Data Utilization:** The use of data by end-users for analysis.

**Feature complexity:** The number of contributing features across a set of intersecting filters.

**Filter Union Data Support Score:** The data support of the data subset that is generated by the union of one or more informative data filters which results in a composite union filter.

**Filter Union Mutual Information Score:** The mutual information of the data subset that is generated by the union of one or more informative data filters that results in a composite union filter.

**Increment Level for (filter) mutual information threshold:** An increment value used to loop through a range of filter mutual information thresholds ranging from a minimum filter mutual information threshold to a maximum filter mutual information threshold.

**Informative Data Filter**: A combination of features and states where the underlying data cluster consistent with the combination has high mutual information against a target feature.

**Intersection of filters:** The data subset that is common to multiple filters.

**Maximum (filter) mutual information threshold:** A maximum value for the mutual information threshold of a filter used to identify a data cluster present in a data set.

**Minimum (filter) mutual information threshold**: A minimum value for the mutual information threshold of a filter used to identify a data cluster present in a data set.

**Mutual information discontinuity threshold: A** discontinuity threshold in the filter union mutual information score used to identify an optimum filter union.

**Relevant Data Set:** The data set that results from an optimal filter union at the filter mutual information threshold where the change in filter union mutual information score exceeds the mutual information discontinuity threshold.

**Simulation entity:** A self contained component that represents one of the active elements in a simulation process. An example of a simulation entity is an agent that comprises a component of an agent based model. An agent-based model (ABM) is a computational model for simulating the actions and interactions of autonomous individuals in a network, with a view to assessing their effects on the system as a whole.

**Testing Data Set:** The data set that is used to evaluate one or more filters and/or one or models.

**Threshold Data Support level:** A normalized value for the percentage of data present in a data cluster derived from a filter.

**Training Data Set:** The data set that is used to identify one or more filters and/or build one or more models.

**Tuning Data Set:** The data set that is used to optimize a model or set of models by adjustment of model parameters.